E-21-6JY
#1

Memory Technology and Trends for High Performance Computing

Contract #: MDA904-02-C-0441

Contract Institution: **Georgia Institute of Technology**
Project Director: **D. Scott Wills**

Project Report
12 September 2002 – 11 September 2004

This project explored the impact of developing memory technologies on future supercomputers. This activity included both a literature study (see attached whitepaper), plus a more practical exploration of potential memory interfacing techniques using the sponsor recommended HyperTransport interface. The report indicates trends that will affect interconnection network design in future supercomputers.

Related publications during the contract period include:

1. P. G. Sassone and D. S. Wills, **On the Scaling of the Atlas Chip-Scale Multiprocessor**, to appear in *IEEE Transaction on Computers*.
2. P. G. Sassone and D. S. Wills, **Dynamic Strands: Collapsing Speculative Dependence Chains for Reducing Pipeline Communication**, to appear in *IEEE/ACM International Symposium on Microarchitecture*, Portland, OR, December 2004.
3. B. A. Small, A. Shacham, K. Bergman, K. Athikulwongse, C. Hawkins, and D. S. Wills, **Emulation of Realistic Network Traffic Patterns on an Eight-Node Data Vortex Interconnection Network Subsystem**, to appear in *OSA Journal of Optical Networking*.
4. P. G. Sassone and D. S. Wills, On **the Extraction and Analysis of Prevalent Dataflow Patterns**, to appear in *The IEEE 7th Annual Workshop on Workload Characterization (WWC-7)*, 8 pages, Austin, TX, October 2004.
5. H. Kim, D. S. Wills, and L. M. Wills, **Empirical Analysis of Operand Usage and Transport in Multimedia Applications**, in *Proceedings of the 4th IEEE International Workshop on System-on-Chip for Real-Time Applications(IWSOC'04)*, pages 168-171, Banff, Alberta, Canada, July 2004.
6. P. G. Sassone and D.S. Wills, **Multicycle Broadcast Bypass: Too Readily Overlooked**, in *Proceedings of the Workshop on Complexity Effective Design (WCED)*, 5 pages, Munich, Germany, June 2004
7. S. Bunchua, D. S. Wills, and L. M. Wills, **Reducing Operand Transport Complexity of Superscalar Processors using Distributed Register Files**, in *Proceedings of the International Conference on Computer Design (ICCD)*, pages 532-535, San Jose, California, October 2003.
8. L. Codrescu, S. P. Nugent, J. D. Meindl, and D. S. Wills, **Modeling Technology Impact on Cluster Microprocessor Performance**, *IEEE Transactions on VLSI Systems*, 11:(5), pages 909-920, October 2003.

9. T. Taha, and D. S. Wills, **An Instruction Throughput Model of Superscalar Processors**, in *Proceedings of the 14th IEEE International Workshop on Rapid Systems Prototyping (RSP03)*, pages 156-163, San Diego, California, June 2003.

10. L. Wills, T. Taha, L. Baumstark, and S. Wills, **Estimating Potential Parallelism for Platform Retargeting**, in *Proceedings of the 9th Working Conference on Reverse Engineering (WCRE)*, pages 55-64, Richmond, Virginia, October 2002.

Questions on comment on this report and activity should be forwarded to the project director at the following address:

Prof. D. Scott Wills
Electrical & Computer Engineering
Microelectronics Research Center
Georgia Institute of Technology
Atlanta, Georgia 30332-0250
(404) 894-7469
scott.wills@ece.gatech.edu

attached: memory trend whitepaper.

Abstract

Despite evolutionary changes over past decades, the fundamental design of computer systems has led to critical communication bottlenecks. Modern microprocessors spend a disproportionate amount of time waiting on transfers between memory, network, and peripherals. The computing industry has developed various solutions to this issue, most following a theme of serialization. Though parallel transfers of data are speed-limited due to wiring complexity and crosstalk, serial transfers at high frequencies can address both the access time and bandwidth facets of communication performance. This is demonstrated in new standards such as PCI Express, HyperTransport, Rambus, SerialATA, and others. This survey paper explains the issues facing modern off-chip communication and analyzes various commercial solutions.

1. Introduction

In recent decades, certain technological trends have been obvious and unmistakable. The most remarkable is that of transistor integration, which currently allows fabrication facilities to produce near-gigascale integration (GSI) levels. This trend, predicted by Intel co-founder Gordon Moore decades ago, has spawned two major others: dramatic processor performance and memory density increases. Companies can now create single processors performing over one GFLOPS (billion floating point operations per second) and single memory chips with over a billion storage elements.

However, changing trends in computer usage place increasing importance on fast network communication, high performance video systems, and faster disk drives. Today's CPUs are capable of producing and consuming more data than the connections can provide, and the division is growing dramatically [6]. This is due to a number of factors including trace lengths, crosstalk, packaging, and others.

This paper will discuss some of the current industrial trends in computer architecture which affect the off-chip communication situation. Section 2 gives an overview of modern computer organization as background. Section 3 discusses the evolutions in memory technology. Section 4 discusses processor to processor connections. Section 5 analyzes peripheral connection solutions. Section 6 discusses other commercial trends that affect off-chip communication. Finally, Section 7 concludes.

2. System Overview

The current organization of a computer was first formulated by John Von Neumann, J. Presper Eckert, and John Mauchly in 1944 while working on the EDVAC program for the U.S. Army. Though modern texts refer to such a design as a Von Neumann machine, this term underemphasizes the impact of these other two men in its design. For simplicity, we also use this term but recognize the group effort involved.

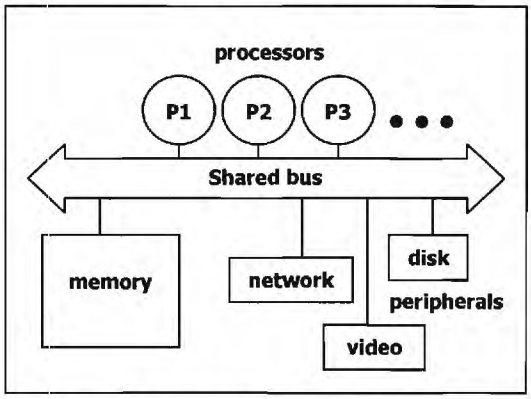


Figure 1. Basic model of computer design formulated by Von Neumann and others. Processor computation and memory density were the bottlenecks when proposed in 1944.

Figure 1 shows the layout of a typical Von Neumann machine with its central communication bus. When this design was developed, the bottlenecks were computation not communication. Thus industrial and academic research went into increasing processor speeds to compute more results and memory density to provide more working space for the computations [6].

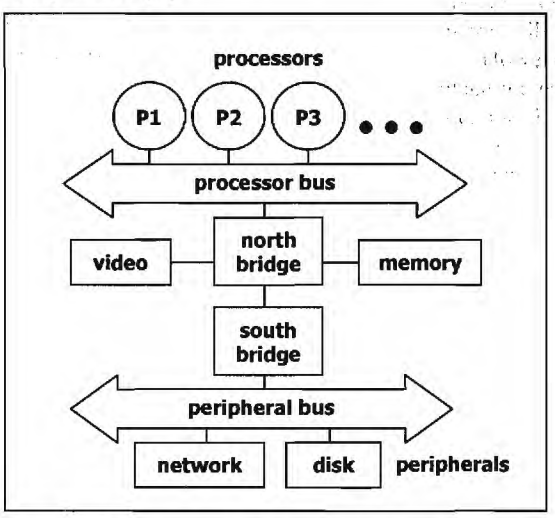


Figure 2. Modern Von Neumann computer design with separate buses connected via bridges. Communication issues are improved in this design, but not completely resolved.

As transistor integration continued to double every 18 months, this single bus became a bottleneck. This central piece could not efficiently arbitrate faster generations of processors and bigger arrays of mem-

ory. Thus the Von Neumann design has slowly been modified over time to accommodate separate connections. These are joined with bridge chips, which arbitrate communication on and between the buses. Figure 2 shows this Von Neumann variation which is used in most contemporary computers.

Though separate buses improved communication pressure, the hunger of modern processors for data from memory only continues to increase. For total computer speed to continue to increase, the use of shared buses must be evaluated.

3. Connections to Memory

The factor of market inertia has kept memory designs fairly uniform. DRAM cells, modules, and signaling are very similar to their counterparts a decade ago. There are some trends, however, which aim to change some of the most fundamental aspects in modern memory.

Memory capacity is, for most applications, not considered a bottleneck. Modern computers contain hundreds, if not thousands of gigabytes of dynamic (non-persistent) memory. More specialized computers contain several terabytes (trillion bytes) of memory [14]. Though few applications require such voluminous memory, these applications are greatly affected by the memory speed. For instance, a Dell Pentium4 2.53 GHz workstation increases its SPECint performance by 5% when moving from the second-fastest available memory to the fastest [16]. This is not "bonus" speedup, but rather performance that was bottlenecked from the processor before the faster memory was installed. It is reasonable to assume that by using even faster memory the bottleneck could be reduced even more (a diminishing marginal return, of course). Clearly modern processors are limited in their performance by the memory subsystem.

Thus academic and commercial research has searched for methods to reduce the three main measures of memory speed: access time, bits/cycle, and cycles/second [2][13]. Rather than quote all three, manufacturers usually quote access time and bandwidth, the latter of which is the product of bits/cycle and cycles/second.

To address these parameters, modern industry has focused on two primary solutions, DDR-DRAM and Rambus. The remainder of this section compares and contrasts these designs. Later in Subsections 6.2 and 6.4, we evaluate emerging solutions (on-chip memory controllers and processor-in-memory) for reducing this bottleneck even further.

3.1 Double Data Rate DRAM (DDR)

The designers of DDR DRAM focused solely on the bits/cycle term. They used the existing and popular SDRAM design and added two phase clocking to transfer data on both the rising and falling edge of the bus clock, essentially doubling the transfer rate without changing clock speeds. The change also requires only small modifications to the module and chip design, increasing its popularity with memory manufacturers [13]. The DDR DRAM standard was ratified by JEDEC, the consortium of memory manufacturers, a few years ago. As such, it is an open standard, thus no royalties are charged for its use. It is currently in full-scale production with no price-premium over its SDRAM counterparts.

This design also continues the positive features of SDRAM design, such as the critical word policy. This is where the module produces the first byte of data very quickly, then follows with the remaining bytes subsequently if necessary. Rambus, as is discussed in the next section, does not implement this performance enhancement [2].

DDR DRAM has its share of disadvantages, however. First, SDRAM was designed before pin counts were an important issue and DDR continues to ignore this factor. Modern DDR DRAM packages have 184 connections, 64 of which are for data. The design favors slow parallel transfers (a large 64 bits per cycle, but slow frequencies of 200 MHz), which shows reasonable performance but at the expense of high pin counts and huge wiring issues on the motherboard. As bus speeds increase, crosstalk and capacitive delays become more significant. As such, DDR in its current form does not scale well into the future [2][13]. The next iterations of DDR, DDR-II and DDR-III, should address these concerns [9].

Performance statistics for DDR-DRAM are shown in Table 1. We also include SDRAM for comparison purposes, though it is not considered a modern memory solution. The first table column lists clock frequency, the second lists transfer rate (product of cycles/sec, bits/cycle, and 0.125 bytes/bit), the third lists access time the critical word, and the fourth lists total pin count. These calculations are for the current top-end (May 2003) modules purchasable.

Table 1. Performance comparison of modern memory designs. The wide and parallel Dual DDR is the best performing, but the serial RDRAM is faster per pin.

	Clock Freq. (MHz)	Bandwidth (gigabytes per sec)	Access Time (ns)	Pin Count
SDRAM	166	1.3	18	168
DDR	200	3.2	12.5	184
Dual DDR	200	6.4	12.5	368
RDRAM	600	4.8	12	162

The table shows that DDR significantly improves on its SDRAM predecessor in both access time and transfer rate. For implementations requiring even higher bandwidth, some designs have incorporated two DDR memory channels. As Table 1 shows, this

doubles the transfer rate but also doubles the (already high) pin count.

3.2 Rambus DRAM (RDRAM)

Rambus is an intellectual property firm with no fabrication facilities of its own. It relies on designing products in-house, then selling ideas to memory and chipset manufacturers. They have produced their own answer to the relative slowing of memory which addresses both parts of the transfer rate formula above.

First, just as with DDR DRAM, the Rambus DRAM module (or RDRAM for short) uses two phase clocking, transferring data twice during a clock period. Their data-bus width, though, is reduced to 16-bits. That would make RDRAMs four times as slow as DDR DRAMs, but Rambus also addresses the cycle time as well. Their modules run at a much higher than usual speed, between 300-600 MHz. For SDRAM or DDR DRAM this would not be possible since the 64-bit data bus is not conducive to high frequencies. But with the smaller RDRAM bus, high frequencies are much easier to implement.

The result is a module capable of up to 2.4 GB/sec with the current clock speeds. This is actually slightly slower than a contemporary DDR module which can transfer up to 3.2 GB/sec. Unlike Dual-DDR, the narrow Rambus channel is explicitly designed for multiple, parallel channels. This allows the bandwidth to be multiplied without producing a prohibitively high pin count. For reference, Intel uses two Rambus channels for its Pentium 4 chipset, the i850 [5]. Rambus modules can also hold a large number of banks open at once, reducing access time to data which has been accessed recently. In addition, the modules include standard power saving modes, increasing their popularity in portable devices. Performance statistics for a typical 2-channel Rambus implementation are shown in Table 1.

The Rambus design has disadvantages, however. First is the ownership of the standard. Since Rambus makes its income from selling its design, each RDRAM module sold includes a royalty for Rambus. The actual amount is fairly small per module, but memory is a low profit margin business. Thus manufacturers are reluctant to invest in producing RDRAM modules, especially when a competing standard like DDR DRAM is royalty-free. Rambus is also in financial difficulty following a succession of court judgments claiming unfair business practices by the management of Rambus. This, too, has not helped their popularity with manufacturers.

The RDRAM chips themselves are also more complex than standard SDRAM or DDR DRAM chips. Though they use the same underlying DRAM layout, the RDRAM design mandates every chip to understand the Rambus protocol (see Figure 3), increasing chip area by around 15%. This added complexity also reduces manufacturing yields significantly, further increasing the price [2]. Currently, RDRAM modules cost over four times more than competing DDR DRAM modules.

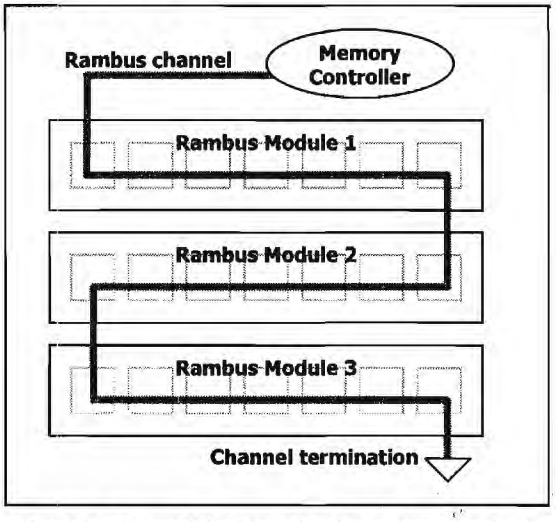


Figure 3. As modules are added to a Rambus channel, the length and the corresponding latency of the channel increases. Also note that each memory chip in the system must understand the communication protocol.

Figure 3 also illustrates an important technical concern about the RDRAM design. The channel is serial, thus adding modules lengthens it and increases latency. DDR DRAM has no such disadvantage since all modules are on a parallel bus, so adding more modules has no affect on access time. Adding capacity by adding *channels* to a Rambus system, however, does not decrease performance as the channels work in parallel. It is only when these channels grow does the signal propagation distance increase.

From a innovation point of view, the Rambus product is superior to the DDR product. Low pin counts mean scalability, power saving modes mean portability, and high bandwidth is essential with modern processor speeds. However, recently Rambus lost their primary partner, Intel, in favor of DDR-DRAM. This leaves their only major customer as Sony, which uses the Rambus modules in their PlayStation products [15]. Thus financial difficulty might force Rambus out of business [2][13]. Their point was well-made, though: serial memory interfaces are scalable and competitive, and are likely to return in future designs.

3.3 Caches

An important aspect of the memory-processor speed discussion is caches. Caches are fast buffers for memory, and are usually located a short distance

away [6]. A first-year computer architecture student would ask why we don't build cache-only systems. Though the question seems naive, the answer is illustrative of cache trends and thus should be answered in some detail.

The student's question is based on the assumption that if some cache is good, then more must be better. This is only partially true. More cache is indeed better, but since caches are complex tables, access times increase with the size of the table. This is not only due to the time to electrically propagate the data in and out of the cache, but also the slowing effect of capacitance that large caches have on their output lines. So, for a cache access time to fit within specification, it can only be so large. As such, designers have been using cache hierarchies for decades now. Smaller, faster caches are placed near the CPU and slower, larger caches are placed closer to memory [6]. Table 2 shows the trend in these hierarchies, which is the movement of bigger caches on-chip. The table also shows that the sizes of these caches are not exponentially increasing, especially considering how quickly main memory size has increased over the same period.

Another interesting trend in this table is how cache dimensions (associativity and line size) haven't changed significantly over time. We speculate this is because the nature of instructions and data haven't changed much over the years. Though there are new instruction arrangements such as single-instruction-multiple-data (SIMD) and very-long-instruction-word (VLIW) making their entrance into mainstream computers, their effect is minimal today. So as long as caches are designed towards optimizing software already on the store shelves, cache dimensions should remain similar to current numbers. When SIMD and VLIW finally become mainstream, we will probably see a small increase in the line size in all cache levels. Data and instructions will no longer be handled sequentially but rather in groups, thus a cache line will accommodate one or more groups.

4. Connections to Other Processors

As Figure 2 shows, the popular choice for computers that have more than one processor is a shared bus design. Design considerations of having more than one processor are discussed in Subsection 6.1, but in this section we discuss commercial improvements for the inter-processor connections themselves. The same motivation that led designers to divide the common bus of Figure 1 into separate buses has also led proposals removing buses altogether. To this end, we will discuss the two relevant inter-processor communication proposals, HyperTransport and Rapid IO.

4.1 HyperTransport

HyperTransport (HT), originally known as Lightning Data Transport (LDT), is AMD's proposal for high-performance interconnections [3]. It addresses the slow and wide nature of modern processor-to-processor connections. These contemporary bus designs use 64 parallel lines which are clocked at a fraction of the processor speed to communicate. This slow frequency is mostly limited by the crosstalk present between these wires.

HyperTransport, on the other hand, is a narrow, high-frequency design. It supports widths of 2 to 32 bits in powers of 2, and supports frequencies of 200MHz to 800Mhz in multiples of 100. The reader should note the similarity to Rambus's design philosophy to replace the slow, wide memory bus with a fast narrow channel. Unlike Rambus, though, transfers occur in sequential packets which are switched to

Table 2. Cache parameters for sample commercial processors. Trends show increasing cache sizes, but relative stability in line sizes and associativity indicating similar target workloads.

			L1 Cache			L2 Cache			
Year	Manuf.	Name	Size (kb)	Line (B)	Assoc.	Size (kb)	Line (B)	Assoc.	OnChip?
1989	Intel	486	8	16	4				no
1991	MIPS	R4000	16	16*	1	8192	64	1	no
1992	DEC	21064	8	32	1				no
1993	Intel	Pentium	32	32	2				no
1994	DEC	21164	8	32*	1	96	32	3	yes
1995	MIPS	R10000	32	32	2	512*	64*	2	no
1998	Intel	Pentium2	32	32	4	512	32	4	no
1998	DEC	21264	64	64	2	2048	32*	3	yes
1999	AMD	Athlon	128	64	2	256	64	16	yes
2001	Intel	Pentium4	16**	6 uops**	4**	512	128	8	yes
2001	DEC	21364	128	64	2	1536	32*	6	yes

* Configurable

** Trace cache used, making parameters non-comparable

their ultimate destination. Traditional processor buses, on the other hand, use one state of address wires and data wires as an entire communication.

The connection is double-pumped so there are two transfers for each clock period. Since crosstalk is not a significant issue with the narrow bus, HT also uses a low-voltage electrical technique called low-swing differential signaling (LVDS). This uses two wires for each data connection: one for the forward data path, and one for the electrical return signal. Low voltage on the pins (+/- 1.2v) results in higher possible frequencies and long maximum trace lengths (up to 24 inches). The obvious cost is that each data wire now requires a corresponding return wire. Thus a 8-bit wide HT connection requires 16 data pins, plus power and ground. Statistics for a typical 8-bit, 800 MHz HyperTransport connection are shown in Table 3.

In addition to not being wide or slow, HyperTransport is also not a bus. As Figure 4 shows, it is a point-to-point connection – there is no common medium for communication. The reader should note, however, that Figure 4 is simplified as that linear processor arrangement has obvious longest-latency issues. Thus real HT systems use geometries such as cubes and hypercubes to reduce longest-distance communication amongst the processors.

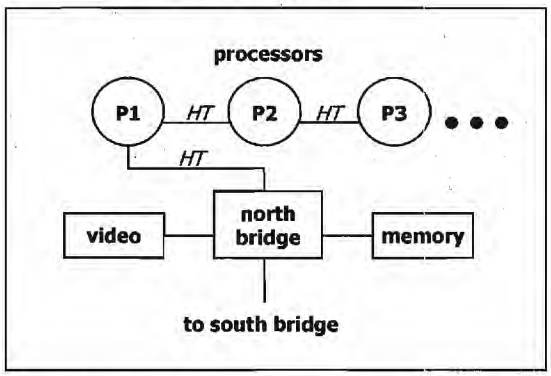


Figure 4. HyperTransport (HT) multiprocessor arrangement which eliminates the common communication medium. This connection geometry (linear) is just an example and is obviously non-ideal.

At the time of this writing, HyperTransport is currently in its infancy. It has only commercially been used for north to south bridge communication. The recent release of the AMD Opteron systems, though, marks the first use of this protocol for inter-processor communication and between the north bridge and the processor. There are over 50 members of its consortium besides AMD and sales of HT-enabled products are expected to exceed 35 million by the end of 2003.

4.2 Rapid IO

Motorola and Mercury Computer Systems have proposed a processor interconnect very similar to HyperTransport called Rapid IO [8]. Similarities include the use of LVDS signaling, double-pumped transfers, packet switching, but most importantly the narrow high-frequency design. Though this connection width is only 8-bits wide, this is actually termed *Parallel* Rapid IO. An even higher-frequency, 1-bit version (*Serial* Rapid IO) is also available for peripheral connections, but we do not discuss it as it is likely to be overshadowed by Intel's offerings discussed in the next section. However, this does bring up interesting possibilities of a complete Rapid IO-based system, with the parallel version connecting the processors and north bridge, and the serial version connecting the peripherals.

Given the similarities of HyperTransport and (parallel) Rapid IO, the HT diagram in Figure 4 is applicable here as well. As before, the shared bus of the Von Neumann architecture has been replaced with point-to-point connections of various geometries. The linear one shown is just an example. The base connection speed is 250 MHz, but 2X (500 MHz) and 4X (1 GHz) versions are also available. Table 3 shows the statistics for the baseline Rapid IO interconnect and the 4X version.

Like HyperTransport, Rapid IO is also in its infancy, but there are already implementations in newer PowerPC-based systems as well as various DSP and FPGA products.

5. Peripheral Connections

Another emerging bottleneck is the peripheral bus used on modern computers. Peripheral Component Interconnect (PCI), introduced by Intel to replace the aging ISA bus, has been the predominant bus for devices such as network cards, mass storage, and USB hubs for over 5 years [7]. The shared bandwidth, however, of a PCI-based system remains 133 MB/sec, which is not sufficient to handle high-bandwidth peripherals such as Gigabit-Ethernet cards or digital video connections. The latest revision to PCI, called PCI-X, increases bandwidth to 1 GB/sec to address these concerns. Though sufficient to handle today's devices, it is not scalable and is expensive to implement because of its high pin count. Thus it only is found in high-end Intel-based servers. Table 3 shows the statistics for existing interconnects, such as ISA, AGP, PCI and PCI-X, as well as for the protocols to be discussed in this section.

Intel's response to the scalability and cost of peripheral bandwidth are two new non-competing 3rd generation I/O connections. In addition, the traditional interconnect to fixed disks (hard drives) is reaching its limit. A newer design called SerialATA

will take its place – another case of serial replacing parallel. Subsection 5.3 discusses this new disk interface design briefly.

5.1 PCI Express

Intel's proposal for third generation I/O is PCI Express. It contrasts PCI and AGP's 32-bit bus with a narrow 'PCI Express Lane' consisting of one wire in each direction. As with HyperTransport and Rapid IO, this smaller bus can run at a much higher frequency through the use of LVDS. Table 3 shows that PCI Express will target an initial speed of 2.5 GHz. With one lane, that produces 312 MB/sec in each direction. More lanes, however, are supported (1, 2, 4, 8, 12, 16, and 32), increasing bandwidth accordingly. Since communication and clocking is simultaneous as each 8-bit byte is encoded with the clock into a 10-bit word, thereby reducing usable bandwidth by 2/10 to 250 MB/sec per lane.

Intel has also chosen to take a packet-routed approach, similar to the network protocol used to run the Internet (TCP/IP). Each PCI Express-based system contains one or more switches, which route packets in much the same way network routers do (see Figure 5). Communication between a peripheral and the system will be bundled in a small packet, then sent (one bit at a time for one lane, n bits at a time for n lanes), to the PCI Express switch which connects this peripheral. This switch then routes the packet based on destination and priority bits to the next switch or destination.

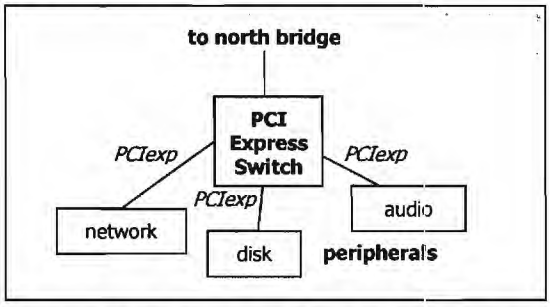


Figure 5. PCI Express switch layout. The traditional peripheral bus (PCI, ISA) has been replaced by a switch, analogous to a router in a packet-based network such as TCP/IP.

Backwards compatibility is one of Intel's primary goals along with performance. Proposals demonstrate the simplicity of adding PCI Express connectors to existing PCI systems, even in the same expansion slots. Thus a slot could be used for a PCI device or a newer PCI Express device. The programming and addressing model for PCI Express devices is also identical to that of PCI, so drivers and operating systems can be completely oblivious to the type of connection this device resides on. Intel expects these factors will allow the transition period from PCI to PCI Express to be much shorter than that of ISA to PCI.

Table 3. Interconnect specifications. Ones in the top section can co-exist in a system with ones on the bottom as they provide a different communication role.

	Clock Freq. (MHz)	Usable Bandwidth (megabytes per sec)	Data Wires	Max Trace Length
Processor Interconnects				
HT	800	1600	16	inches
RapidIO	250	500	16	inches
RapidIO-4X	1000	2000	16	inches
Peripheral Interconnects				
ISA	8	16	16	inches
PCI	33	133	32	inches
PCI-X	133	1066	64	inches
PCI-X 533	533	4266	64	inches
AGP-8X	533	2133	74	inches
PCIexp	2500	250	2	yards
PCIexp-4X	2500	1000	8	yards
Infnbd	2500	250	2	miles
Infnbd-12X	2500	3000	24	miles
Disk Interconnects				
Parallel ATA 66		133	16	inches
UW-SCSI-3	40	160	16	yards
SerialATA	1500	150	4	yards
SerialATA-4X	6000	600	4	yards

The final advantage of PCI Express is cost. Since the interconnect is by-design narrow, less traces and connections are necessary than with PCI. As Table 3 shows, traditional PCI has 32 data pins and PCI-X has 64, but PCI Express has only 2 per lane. Even operating at a higher frequency, these connections should be no more expensive than implementing PCI on a motherboard [5].

In addition to its technical advantages over PCI and AGP, Intel's powerful influence in the computer marketplace has guaranteed this standard will be adopted and will succeed. Motherboards and devices are expected to be released by the second half of 2003, but there is little news on the common number of lanes expected (and thus the expected bandwidth).

5.2 Infiniband

Intel's other solution to scalable bandwidth comes in the form of Infiniband [4]. The distinction from PCI Express comes in the objective of Infiniband -- a unified I/O fabric. Here, Intel wishes to consolidate the many interconnects used below the south bridge – ATA, SCSI, Ethernet, Fibre Channel, etc. – into a single communication protocol. This solution greatly resembles PCI Express with a narrow, high-

frequency design enabling high bandwidth and long maximum trace-lengths.

Since mainstream users are unlikely to replace their Ethernet network infrastructure or ATA disks, Infiniband aims at the data center market where the unified model of devices is practical and affordable. An Infiniband system is unique because it treats disks, printers, and computers as simple nodes on a network. This equality gives programmers and IT professionals greater simplicity in designing applications in these clustered environments.

From a physical point of view, however, Infiniband looks very similar to PCI Express: it uses single-bit-pair channels with LVDS, it uses multiple channels to scale bandwidth, it runs at 2.5 GHz, and communication is 10/8 bit encoded and packet switched. Table 3 shows specifications for a single-channel Infiniband system and a 12-channel one. The reader should note from the table that the maximum trace-length is several miles, significantly higher than that of PCI Express.

Given the different target markets, both PCI Express and Infiniband will co-exist in Intel's roadmaps for years to come. Infiniband devices are currently in low production currently, so can only be purchased at significant cost.

5.3 SerialATA

The traditional interconnects for disk drives are parallel ATA (also known as IDE) and SCSI. These are both wide, low-frequency connections which are reaching their scalability limits. Though magnetic disk drives have not dramatically increased in speeds (relative to processors), they will reach the limit of these protocols shortly.

The proposed replacement for parallel ATA and SCSI comes in the form of SerialATA [10]. Similar to previously discussed replacements for parallel interconnects, SerialATA is a narrow high-frequency design. It uses two LVDS bit-pairs (one pair for transmit, one for receive) which use 10/8 bit encoding to transmit the clock. The frequency of the initial SerialATA specification is 1.5 GHz, producing a maximum usable bandwidth of 150 MB/sec (not counting the 20% due to clock encoding). This is barely more than the best ATA bandwidth of 133 MB/sec or the 160 MB/sec of ultra-wide SCSI-3. However, the scalability of the serial design will allow at least 2X and 4X versions moving speeds to 600 MB/sec.

Unlike PCI Express, SerialATA is not packet-based and does not allow for switches. Multiple SerialATA controllers, however, could be switched by a PCI Express controller. Numbers for existing disk connections and SerialATA are included in Table 3 at the bottom.

SerialATA (1X) devices (disks and controllers) are currently in the marketplace at relative price parity with their parallel ATA cousins. Given the limited speed of disks, little performance is gained by the transition currently. The industry believes, however, that moving to the scalable serial platform now is wise to avoid future bottlenecks.

6. Other Relevant Developments

There are many ideas in commercial architecture which may become mainstream within the next decade. This section evaluates a few of the more important ones relevant to off-chip communication issues.

6.1 Chip Multiprocessors (CMP)

Multiprocessors, systems with multiple processors, have been around since the first supercomputers were constructed. Conventional multiprocessing puts all of the processors on a circuit board and lets them communicate through a type of off-chip interconnect (traditional bus, HyperTransport, etc.). The traces that make up this board-level connection, though, are many centimeters long and therefore must be clocked conservatively.

A chip multiprocessor, on the other hand, puts these multiple processors on the same chip. This integration eliminates the off-chip communication between processors. These traces are obviously much smaller (width and length) than those connecting separate chips in conventional multiprocessing, so communication channels can be significantly accelerated.

Cost savings are also a potential advantage of CMPs as only one chip is packaged and the motherboard design can be simplified. In addition, the chip multiprocessor is not significantly harder to design than a single processor. A CMP simply contains multiple copies of that processor design plus communication logic. However, some of these savings are offset by the reduced yield of larger chips.

It is consumer usage patterns, not yield numbers though, that keep CMPs from mainstream computing. A multiprocessor (chip-scale or otherwise) is only useful when there is more than one thing to do at once. This is why, traditionally, multiprocessors are used in servers and datacenters, where there are many transactions and jobs to be divided amongst the processors. End users, on the other hand, rarely do more than one computing-intensive task (games, video encoding, complex spreadsheet computations, etc.) simultaneously. Single-task usage patterns on a CMP would leave all processors but the first idle, unable to assist in speeding up the user's processing. Thus, a better use of this chip's transistors would have been a larger, more complex single processor.

However, CMPs may become more mainstream as applications become more *multithreaded* in nature.

Threads are pieces of a program that are relatively independent of other parts of the program. When used effectively by a programmer, threads allow a single program to be divided and conquered by an MP system. Though multithreading has been around for many years, the increasing popularity of thread-friendly languages such as Java and C++ should mean more applications will be better threaded as time continues.

6.2 On-die Memory Controller

Another industry solution to memory latency has been to eliminate the north bridge as the middle-man. AMD, in their new Opteron processor, has integrated the memory controller on to the die [1]. This reduces memory latency significantly, though actual numbers are not publicized. In addition, multiple Opteron systems do not share main memory between CPUs. Each processor gets a full memory channel and thus memory bandwidth scales linearly with the number of processors present.

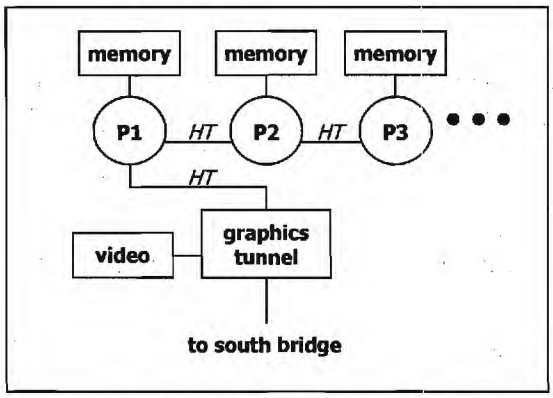


Figure 6. System with on-die memory controllers, such as the AMD Opteron. With inter-processor and memory communication moved out of the north bridge, its only remaining functionality is communication to the video card.

Interestingly, since Opteron systems also employ HyperTransport for inter-processor communication, this reduces the north bridge responsibilities to only one – access to the video card. Thus Opteron chipsets use the term "graphics tunnel" to describe the north bridge chip which no longer bears most of the normal responsibilities of such (see Figure 6).

6.3 North Bridge Ethernet

Though Intel has yet to move their memory controller on-die, they have promoted one of the most important aspects of a modern workstation to the north bridge – a dedicated Ethernet bus called Communications Streaming Architecture (CSA) [11]. Given the increasing use of local-area networks and the bandwidth crunch of modern (pre-PCI Express) systems, Intel has developed a new north bridge which puts network communication on par with video and memory traffic (see Figure 7).

Performance gains are quite tangible as a gigabit Ethernet controller on a CSA bus transfers data 69% faster than its PCI counterpart. This speed is of limited practicality, though, as disk drives are not capable of reading or writing data that quickly. So data transferred at that speed would have to be immediately used (i.e., streaming video) to see speedup over the PCI version.

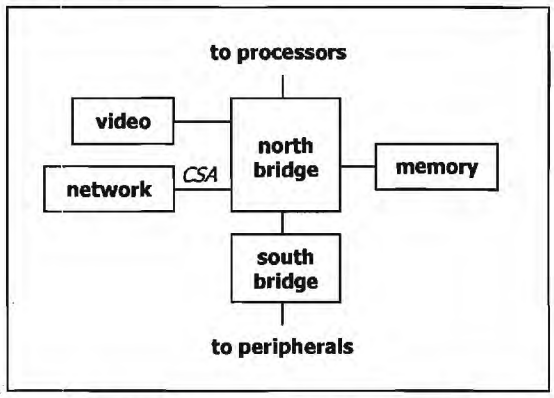


Figure 7. Intel Communication Streaming Architecture (CSA) layout. Here the network connection (gigabit Ethernet) is promoted to the north bridge to remove bottlenecks.

CSA is very interesting because of its promotion of the network controller – further evidence of the communication over computation trend in commercial computing. The long term future of the CSA bus is questionable, though, as PCI Express should handle gigabit network devices with ease.

6.4 Processor in Memory (PIM)

Processors-in-memory directly attack the relative slowing of memory by proposing a change to the most basic of computing paradigms. Instead of a bus-centric system, some researchers propose a memory-centric system. In PIM-based machines, the processor becomes merely a collection of functional units attached to main memory, and peripherals become the communication mechanisms for it (see Figure 8). A less radical version of this proposal is the addition of a few computational features to memory chips. These processor-in-memory units completely subvert the load/store that a typical Von Neumann system would need and do entire operations within memory, reducing traffic on any exterior buses. Either mechanism, however, would involve changes to the most basic of programming paradigms. Existing code would have to be drastically modified to execute on a PIM machine, making wide-scale adoption unlikely [12].

This idea faces a another major hurdle: memory-logic integration. These two chip industries have

evolved in different directions over the years. Memory gates, for instance, are vertical, not side-to-side or up-and-down like logic gates (this is done to increase the density of very regular layouts such as DRAM cells) [12]. Very few companies attempt to fabricate both DRAM and logic chips separately, let alone in the same chip. In short, any successful PIM designer will have to create their own infrastructure to fabricate these hybrid chips.

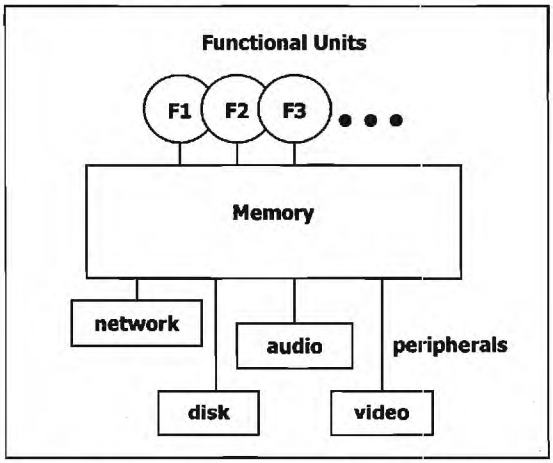


Figure 8. Processor-in-memory (PIM) architecture. Centric memory is served by functional units for computation and peripherals for communication, requiring different programming paradigms.

If any area of computing seems a likely recipient of PIM ideas, graphics cards are a good bet. Modern video chips frequently perform regular operations to whole regions of memory (darken, anti-alias, rotate, etc.) and are extremely memory-bandwidth bound. Functional memory could be very beneficial to performance, and the only programming that would have to be modified is the video driver.

7. Conclusions

Clearly there exists no one solution to the limitations of the Von Neumann architecture. In fact, most companies and trade associations focus more on improving their products individually than on improving how the system works as a whole. However, many conclusions can be drawn from the trends of commercial solutions to interconnect issues.

The primary conclusion is that low pin count, high frequency interconnects are inevitable. Evidence towards this conclusion can be found in newer interconnects such as HyperTransport, PCI-Express, SerialATA, USB, FireWire, and so on, which are all high-frequency serial replacements for parallel ancestors (ATA, parallel ports, etc.). High frequency serial connections give many advantages such as less wiring, lower voltage, longer traces, and ultimately higher performance with greater scalability. Many of these techniques are non-competitive and can co-exist in a computer, such as the example system in Figure 9.

A second conclusion is that cache will be become even more important. It is unlikely that the memory speed gap will shrink in the near future, and thus cache is still the quick-fix for the issue. As was discussed earlier however, cache sizes cannot increase dramatically, so the memory performance issues must be continually addressed.

Finally, integration of devices closer to the processor will continue to increase. Moving memory controllers on to the processor and moving network cards to the north bridge are only the beginning. The culmination of this trend is vaguely referred to as system-on-a-chip (SoC), when all devices (CPU, chipset, memory, etc.) are integrated on a single wafer. This may lead to the use radical computer designs such as processor-in-memory, but legacy applications will be a major hurdle. Practically, fabrication costs will keep any organization of SoC in the future for several years to come.

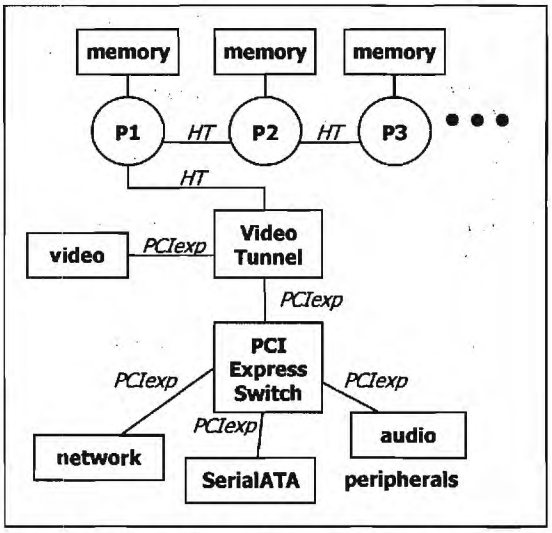


Figure 9. Possible use of HyperTransport (HT), PCI Express (PCIexp), and Serial ATA in a single system. All buses have been successfully replaced with high frequency serial connections.

References

[1] AMD Corporation, http://www.amd.com, last accessed: May, 2003.

[2] Johan De Gulas. "Ace's Guide to Memory Technology, Parts 1 and 2," Ace's Hardware, http://www.aceshardware.com/ Spades/read.php?article_id=5000173., last accessed: May 2003.

[3] HyperTransport Consortium, http://www.hypertransport.org, last accessed: May 2003.

[4] Infiniband Trade Association, http://www.infinibandta.org, last accessed: May 2003.

[5] Intel Corporation, http://www.intel.com, last accessed: May 2003.

[6] David Patterson and John Hennessy, *Computer Architecture: A Quantitative Approach,* Morgan Kaufmann, 1995.

[7] PCI Special Interest Group (PCI-SIG), http://www.pcisig.com, last accessed: May 2003.

[8] Rapid IO Trade Association, http://www.rapidio.org, last accessed: May 2003.

[9] Jack Robertson, "JEDEC Solidifies DDR-II Specification," EBN News, http://www.ebnews.com/digest/story/OEG20010625S0104, last accessed: May 2003.

[10] SerialATA Working Group, http://www.serialata.org, last accessed: May 2003.

[11] Anand Lai Shimpi, "CSA," AnandTech.com, http://www.anandtech.com/chipsets/showdoc.html?i=1811&p=5, last accessed: May 2003.

[12] Jurij Šilc, Bourut Robič, and Theo Ungerer, *Processor Architecture*, Springer Publishing, 1999.

[13] Jon Stokes, Ars Technica, http://arstechnica.com/paedia/r/ram_guide/ram_guide.part3-1.html, last accessed: May 2003.

[14] NEC Earth Simulator, http://www.es.jamstec.go.jp/esc/eng, last accessed: May 2003.

[15] Sony Playstation Hardware, http://us.playstation.com/ hardware/ PS2/415007657.asp, last accessed: May 2003.

[16] Standard Performance Evaluation Corporation (SPEC), "Benchmark Result Search Engine," http://www.spec.org, last accessed: May 2003.

Commercial Memory and Northbridge Technology

Peter Sassone

Georgia Institute of Technology

Introduction

- There are many real-world architectural issues in off-chip communication:
  - Between processors
  - Between peripherals
  - Between computers
- The primary one that really concerns architects is the infamous processor-memory speed gap.
- There are many commercial solutions to the problem, though all are quite conservative.

Introduction

- Industry is ruled by one thing: profit.
- Difficult to make a profit being *revolutionary*, so companies choose *evolutionary*.
- Thus "legacy compatibility" is requirement number one in any marketplace solution.
- So commercial solutions often strive to achieve performance *and* backwards compatibility.
- One of the most important of which is the changing role of the real "central processing unit". *Hint: it's not the processor.*

Outline

- Memory Technology
  - SRAM
  - DRAM
- Commercial Offerings
  - FPM DRAM / EDO DRAM
  - SDRAM / DDR SDRAM
  - RDRAM / XDR
  - DDR 2 / GDDR 3
- Northbridge
  - Old role
  - New role

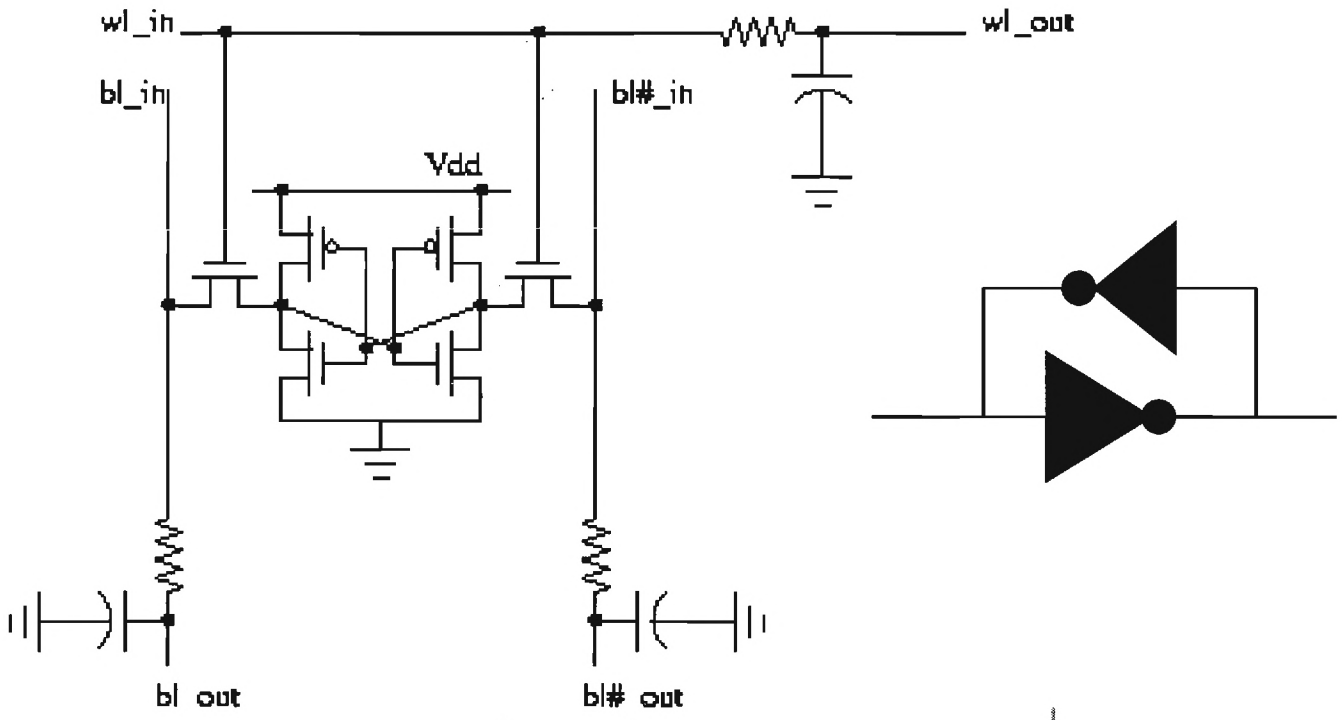
Cache History

- No discussion of memory is complete without a look at caches.
- Lots of interesing trends here, but the most important is the increasing sizes. Why?

			L1 Data Cache			L2 Onchip Cache		
Year	**Manuf.**	**Name**	**Size (KB)**	**Line (B)**	**Assoc.**	**Size (KB)**	**Line (B)**	**Assoc.**
1989	Intel	486	8	16	4			
1993	Intel	Pentium	32	32	2			
1994	DEC	21164	8	32*	1			
1995	MIPS	R10000	32	32	2			
1998	Intel	Pentium2	32	32	4			
1998	DEC	21264	64	64	2			
1999	AMD	Athlon	64	64	2	256	64	16
2001	Intel	Pentium4	8	64	4	512	128	8
2001	DEC	21364	64	64	2	1536	32*	6
2003	AMD	Opteron	64	64	2	1024	64	16
2003	Transmeta	Efficion	64	64	4	1024	128	4

SRAM

- SRAM : static random access memory
- SRAM is a misnomer: it is very dynamic!
- It is constantly inverting and un-inverting a single bit.



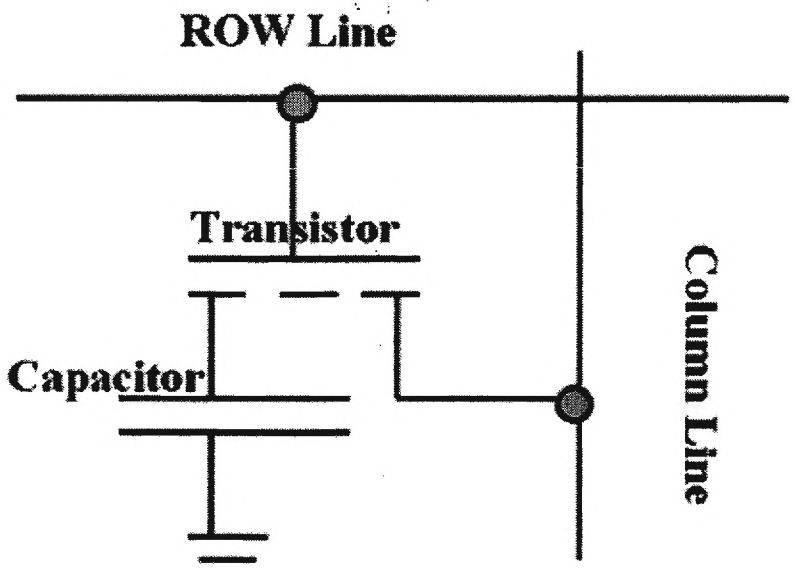
Georgia Institute of Technology

SRAM Commentary

- Quite fast
  - P4 L1 has single cycle access!
- Quite leaky
  - Fast SRAM == lots of leakage
- Quite big
  - 6 transistors to store 1 bit
  - DRAM can name that song in 1 transistor
- Best use: caches and registers
  - SRAM is easy to lay out, and in small amounts is not too power hungry, especially with tricks (sleeping).
  - Pentium M has a 1MB L2… it must have a good performance/power tradeoff!

DRAM

- DRAM : Dynamic Random Access Memory
- DRAM is dynamic for a different reason than SRAM
- Value stored with substrate capacitance, but value must be 'refreshed' every so often or the capacitance dissipates.

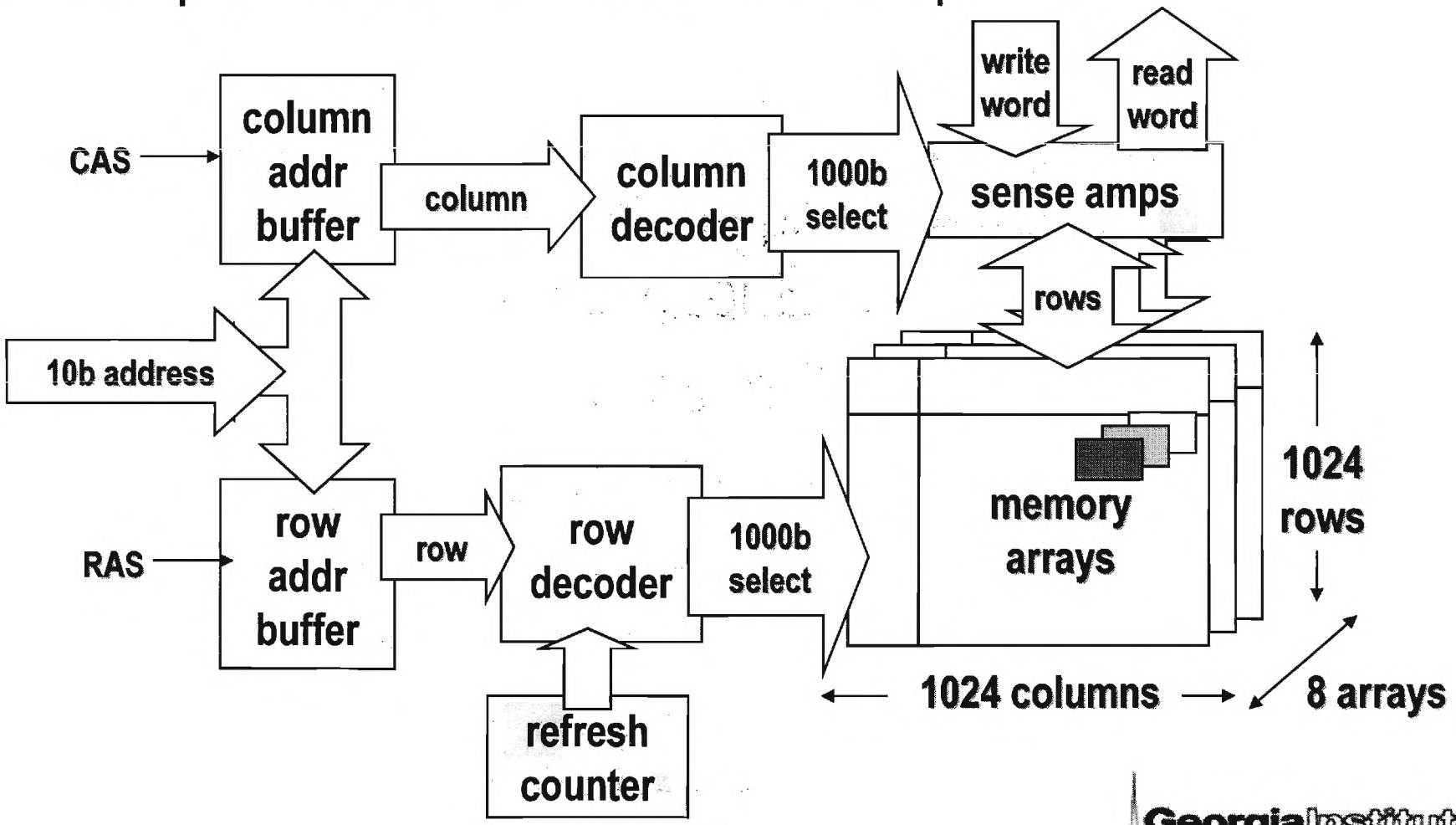


DRAM Operation

- A DRAM cell is very low power
  - C is on the order of 1 10^{-15} F == 40,000 electrons.
- Such low power that alpha particles cause DRAM soft errors (thus the use for ECC).
- For all practical purposes, a DRAM is an analog device and thus is sensitive to power and noise margins.
- It is complicated to read and write to a DRAM cell, but lets try anyway.

DRAM Operation

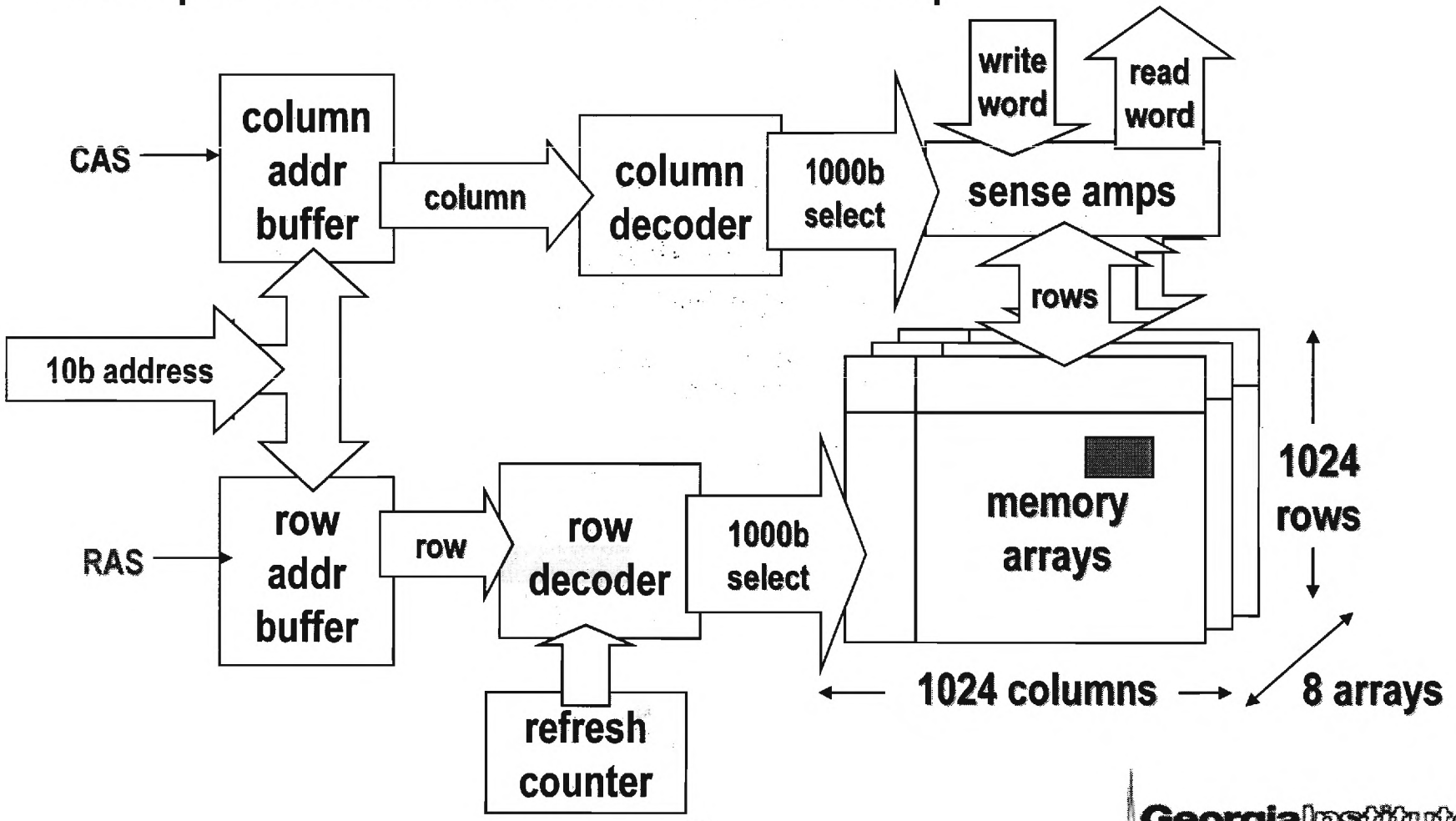
- Sample read on a 1MB DRAM chip:



Georgia Institute of Technology

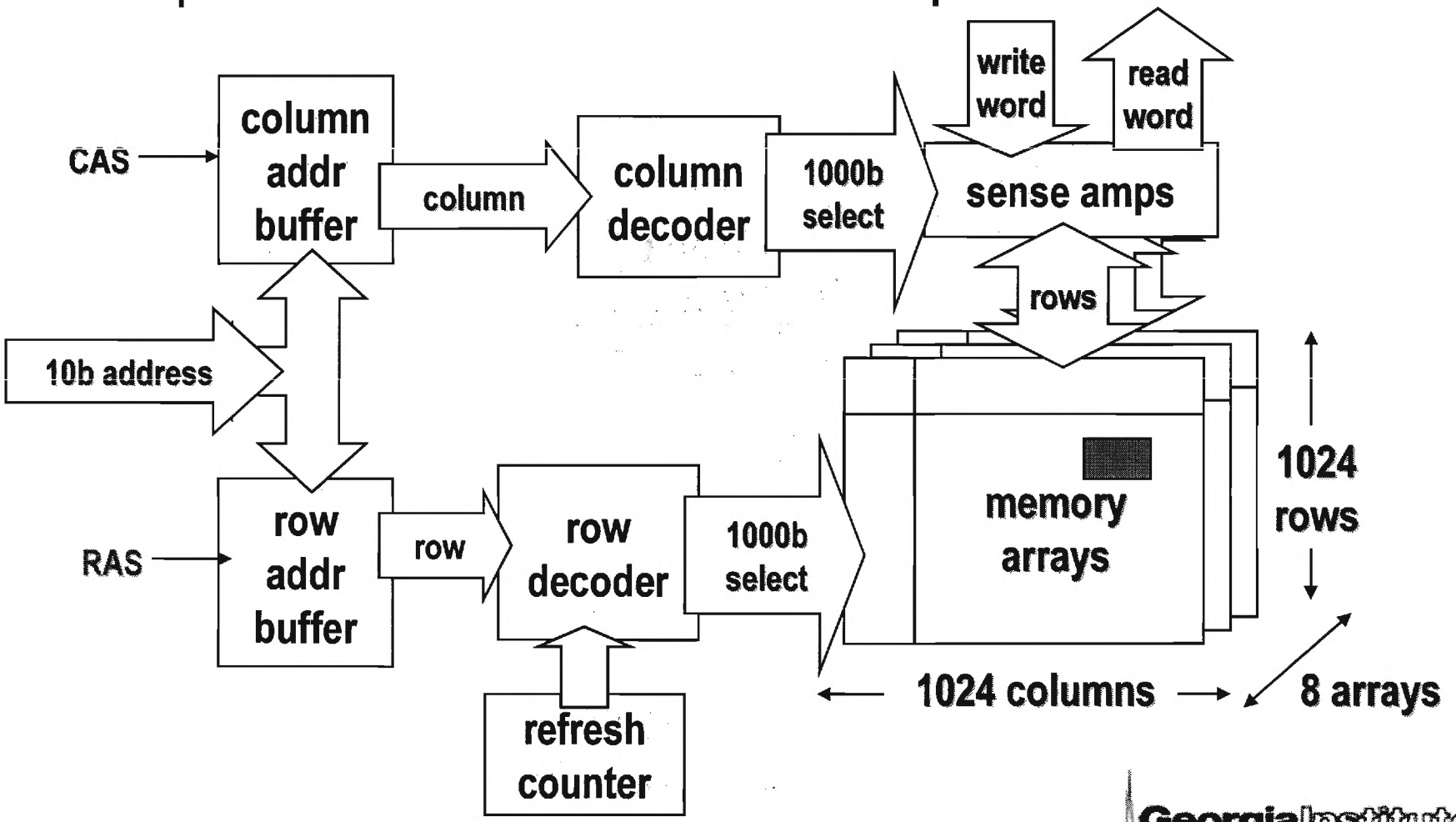
DRAM Operation

- Sample read on a 1MB DRAM chip:



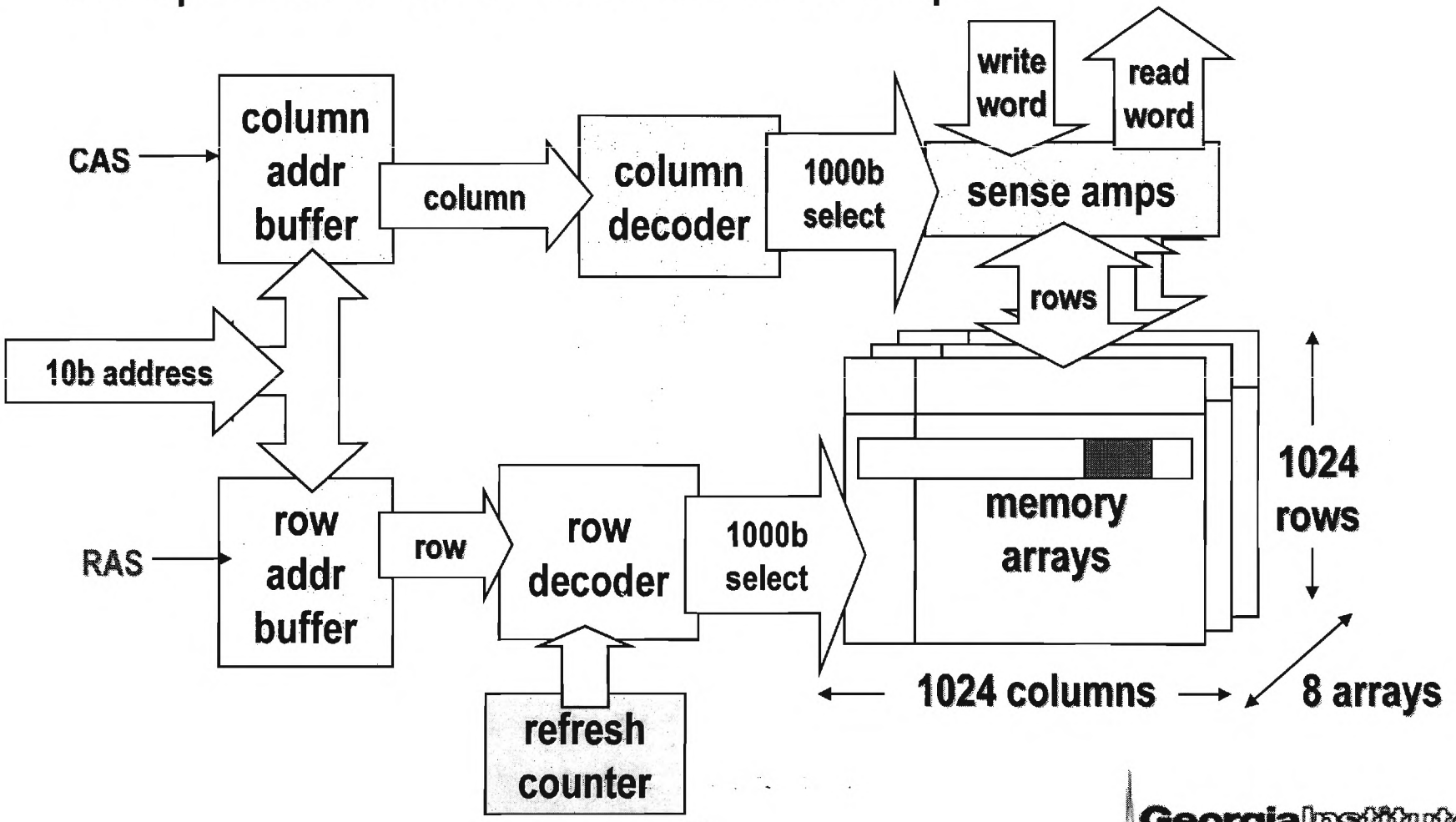
DRAM Operation

- Sample read on a 1MB DRAM chip:



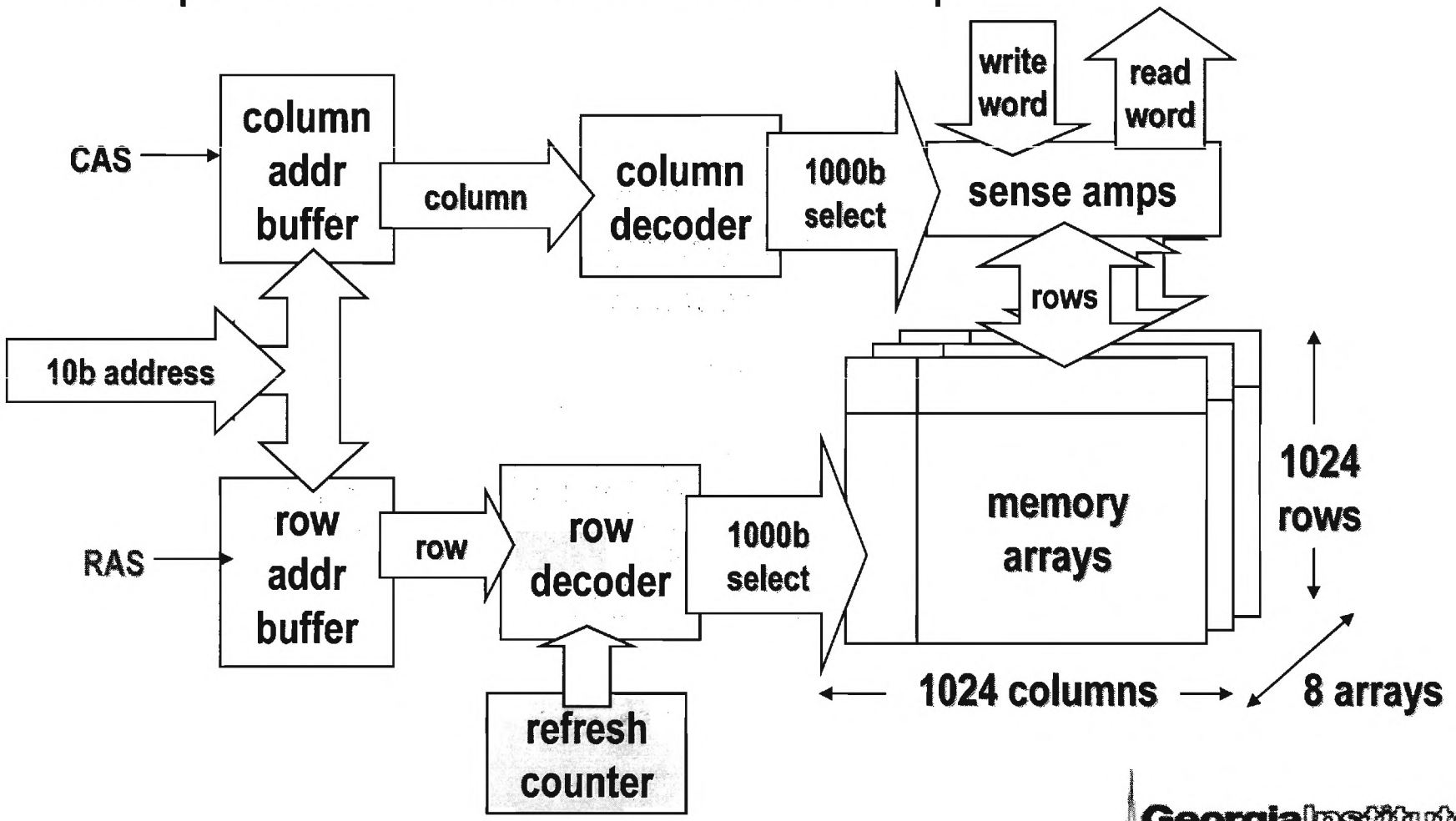
DRAM Operation

- Sample read on a 1MB DRAM chip:



DRAM Operation

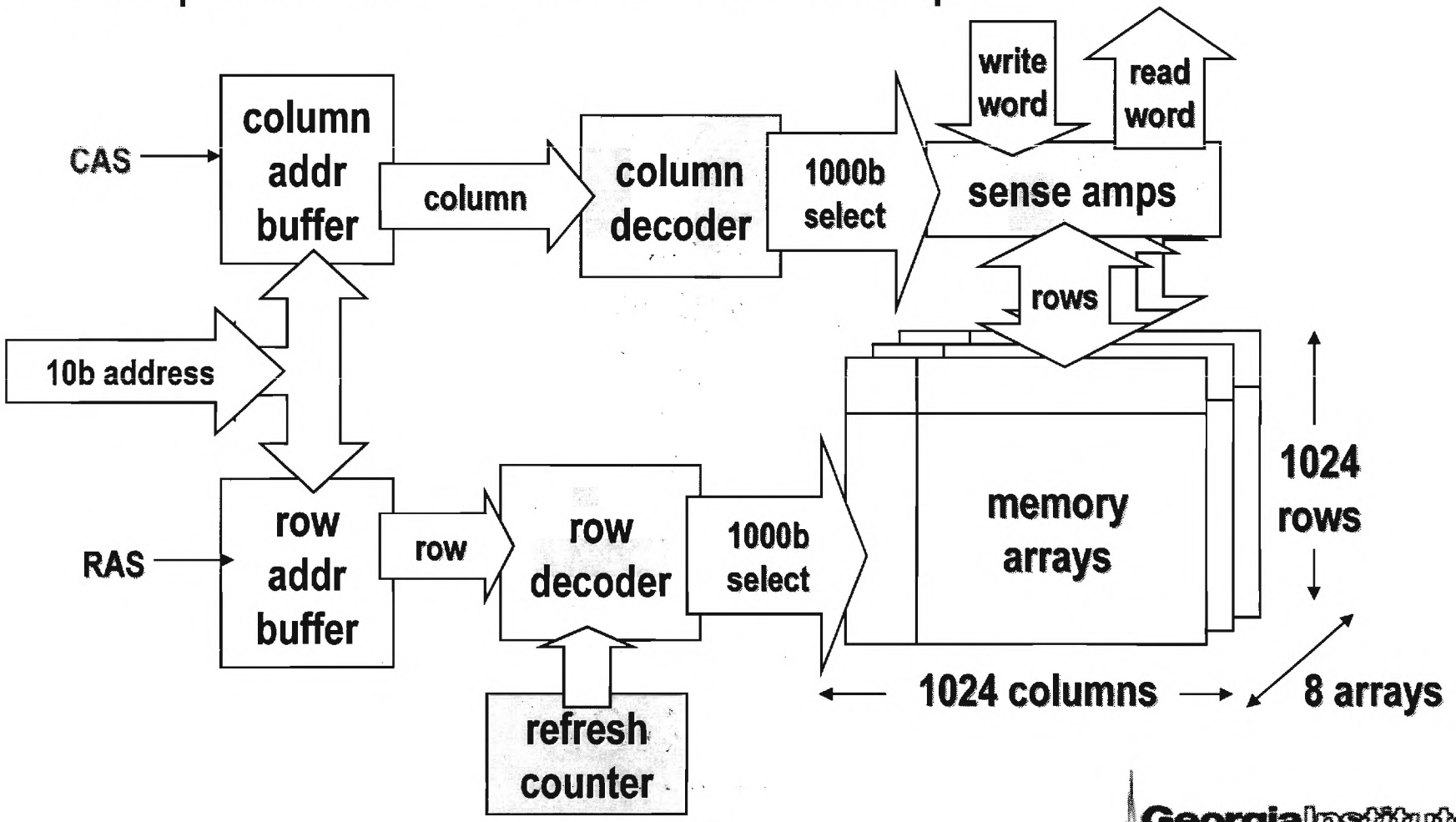
- Sample read on a 1MB DRAM chip:



Georgia Institute of Technology

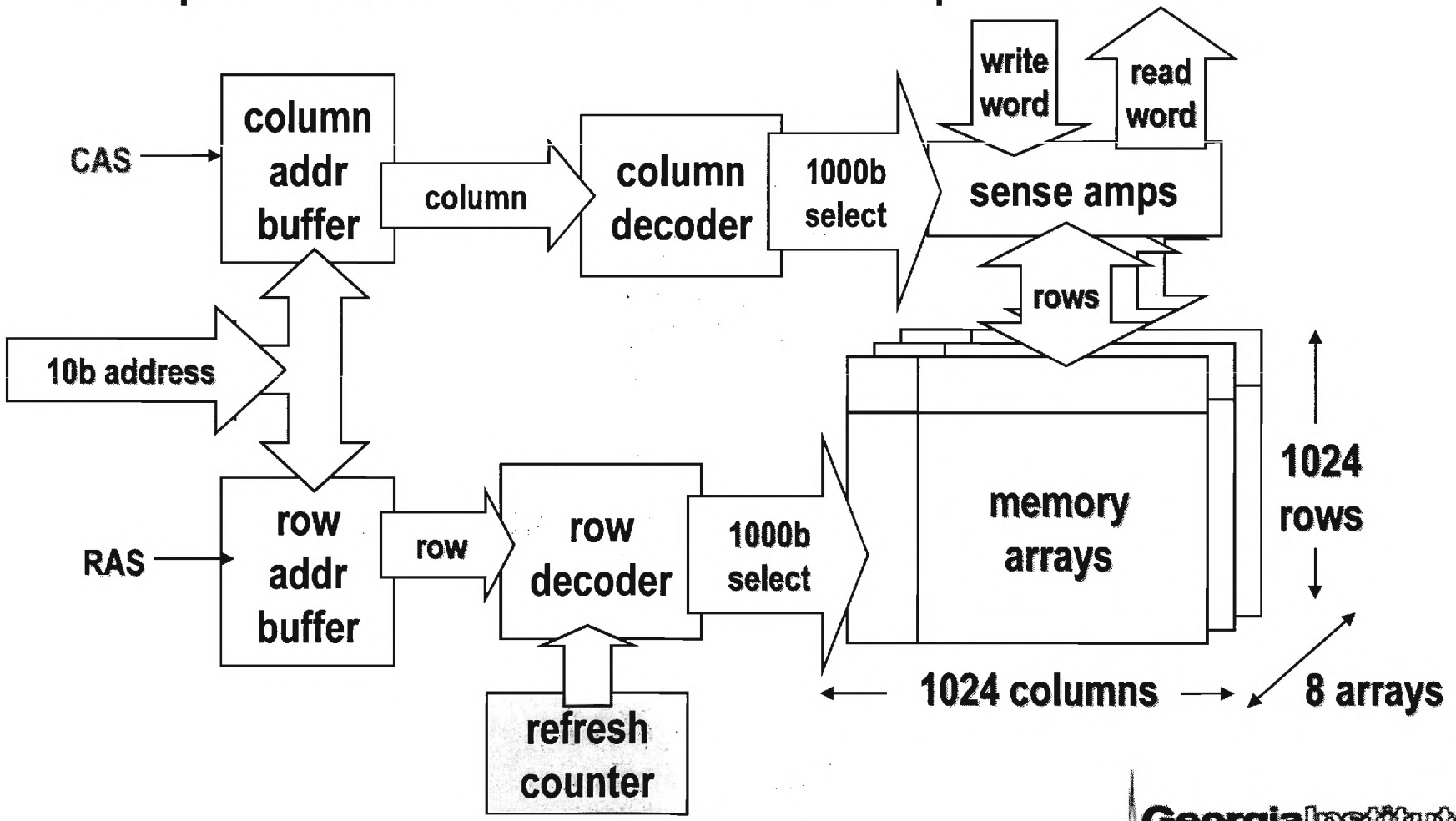
DRAM Operation

- Sample read on a 1MB DRAM chip:



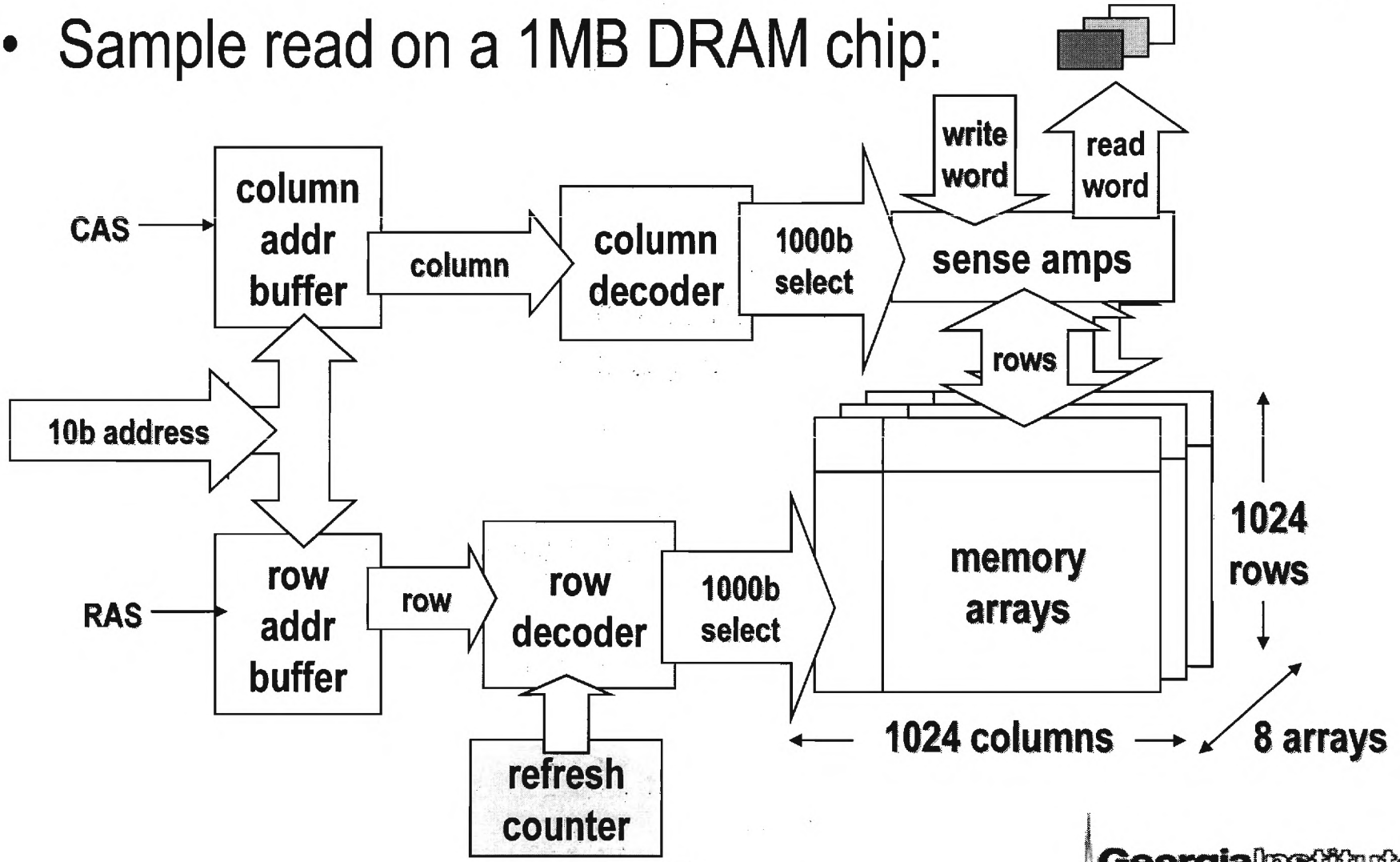
DRAM Operation

- Sample read on a 1MB DRAM chip:



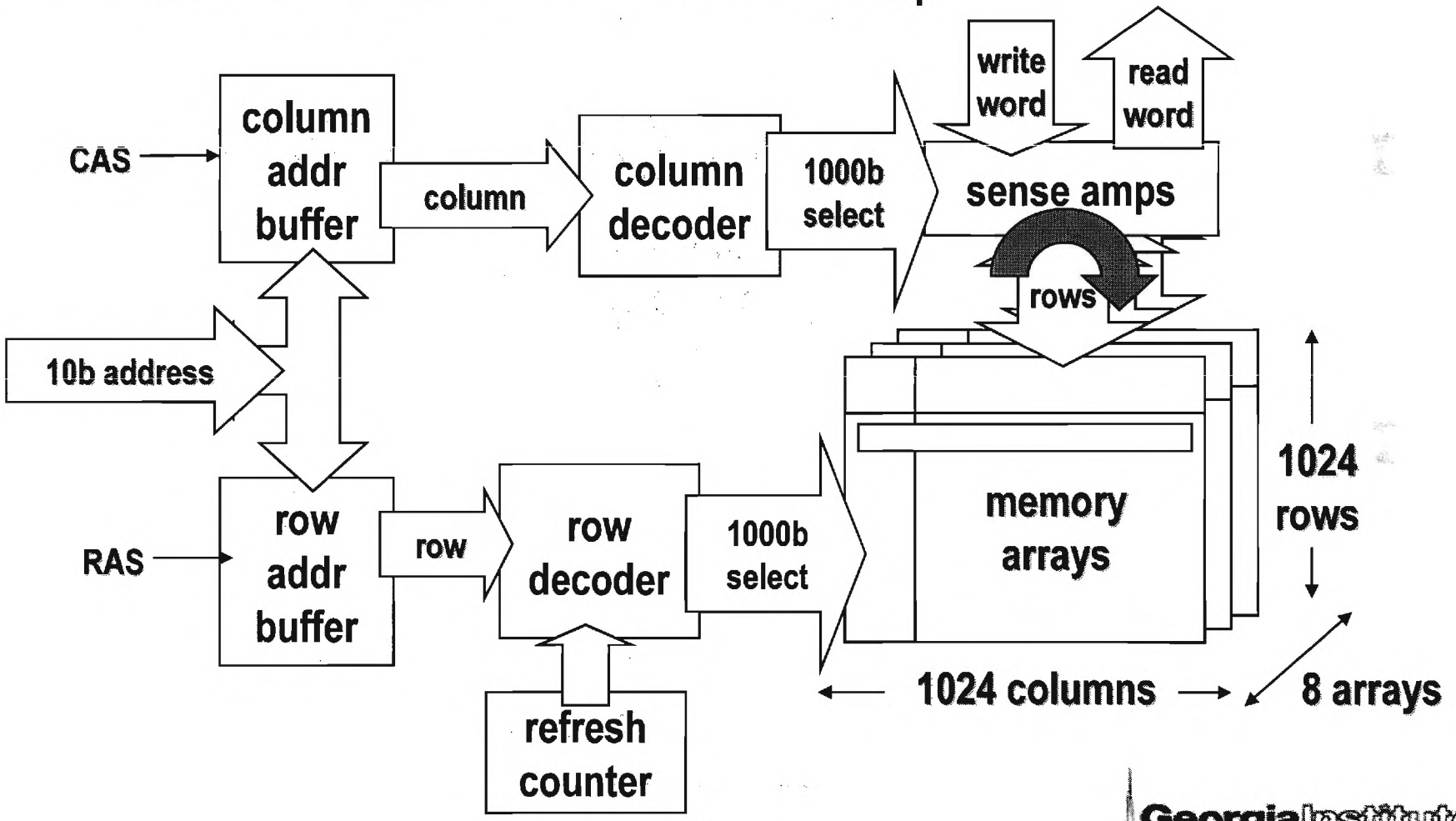
DRAM Operation

- Sample read on a 1MB DRAM chip:



DRAM Operation

- Row refresh on a 1MB DRAM chip:



Georgia Institute of Technology

DRAM Parameters

- Your PC's BIOS is filled with DRAM parameters:
  - CAS Latency – cycles for column decoder to work
  - RAS-to-CAS Delay – cycles between changing from row address to column address mode
  - RAS Precharge – cycles for sense amps to detect a row
  - Recharge Delay – Cycles between row refreshes. Must be no greater than 15us or so.
    - Recharging 1000 rows takes 15us * 1000 = 15ms!
    - Row is inaccessible during this time

Outline

- Memory Technology
  - SRAM
  - DRAM
- Commercial Offerings
  - FPM DRAM / EDO DRAM
  - SDRAM / DDR SDRAM
  - RDRAM / XDR
  - DDR 2 / GDDR 3
- Northbridge
  - Old role
  - New role

Commercial Offerings

- As we said before, industry chooses compatibility for economic (profit) reasons.
  - Compatibility with other devices
  - Compatibility with infrastructure
- All *DRAM variations are basically just regular DRAM (called Fast Page Mode, FPM) with minor modifications.
  - These can increase performance significantly.
  - Don't change the way that DRAM is manufactured.
  - Don't change the paradigm for processor-memory communication.

EDO DRAM

- EDO DRAM: Extended Data Out DRAM
- Innovation: Pipelined accesses.
  - A column select from an access can occur while the row select from the next access is occurring.
  - Effective bandwidth improvement over FPM is about 30%, though max bandwidth is the same.
- Chips sold worked at 66Mhz, reading 32 bits at once:
  - 66 Mtrans/sec * 4 B/trans = 266 MB/sec

SDRAM

- SDRAM : Synchronous DRAM
- Innovation: synchronous interface to the system bus
  - Chip itself no more synchronous than EDO or FPM.
  - All DRAM is clocked.
- Came in a large variety of clock speed grades, but all were 64bit wide.
- Intel created standards to rate memory speed
  - PC133 : 133 Mtrans/sec * 8 B/trans = **1066 MB/sec**
  - Rating is "PC" + the clock frequency of the module.

DDR SDRAM

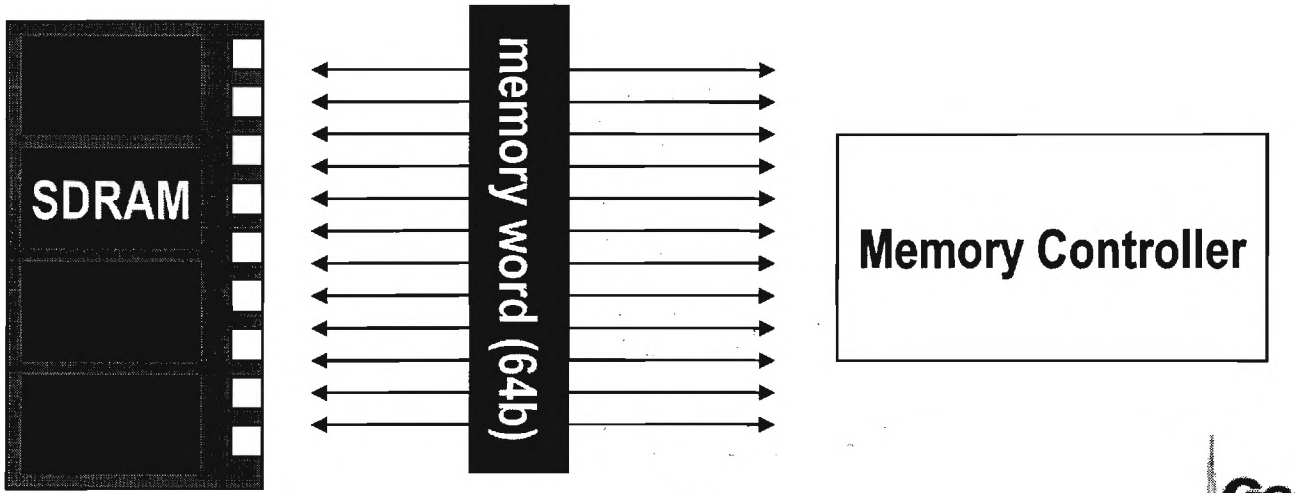
- DDR SDRAM : Double Data Rate SDRAM
- Innovation : Transfer words on both the rising and falling edges of the clock.
  - RAS and CAS lookups happen at the same rate
  - Only the movement of data out of the sense amps is increased.
- Intel created rating systems here too:
  - PC2100 : 133 Mcyc/sec * 2trans/cyc * 8 B/trans = **2133 MB/sec**
  - PC4200 : 266 Mcyc/sec * 2 trans/cyc * 8 B/trans = **4267 MB/sec**
  - Rating is "PC" + approx bandwidth in MB/sec

Recap and Problems

- Things have come a long way since EDO:
  - EDO 0.266 GB/s
  - SDRAM 1.066 GB/s
  - DDR 2.133 GB/s
- However, still wasn't fast enough to satiate CPUs.
  - Definitely not enough for GFX cards (more later).
  - Need more BW **and** less latency
- Trace density was getting bad.
  - Two channels of DDR > 350 traces!

RDRAM

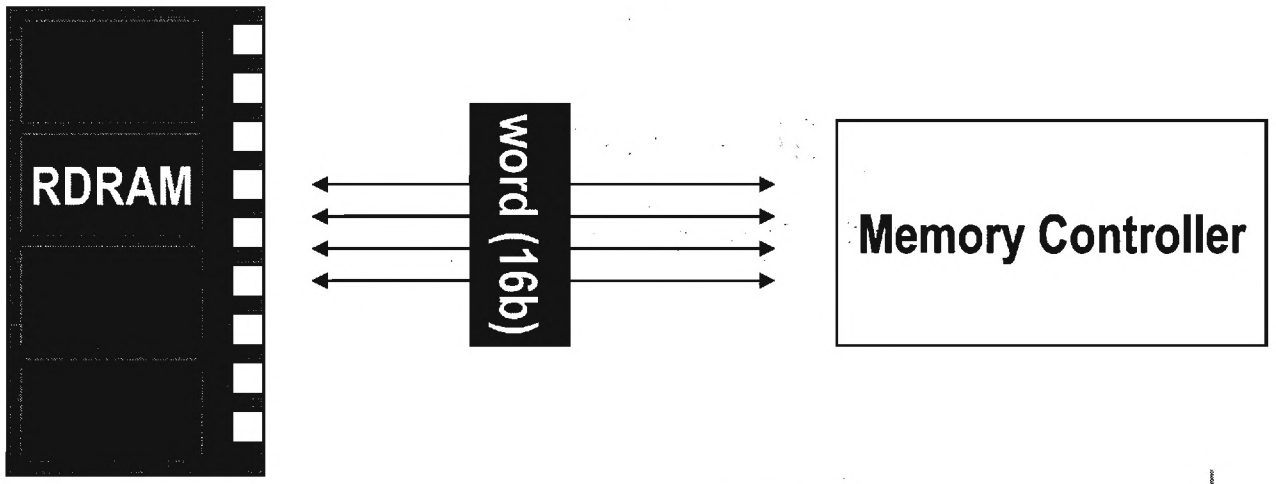
- RDRAM : Rambus DRAM
- Innovation : Reduce crosstalk problems by clocking a narrower memory channel faster.
- SDRAM (DDR included) transfers data in 64bit chunks:



Georgia Institute of Technology

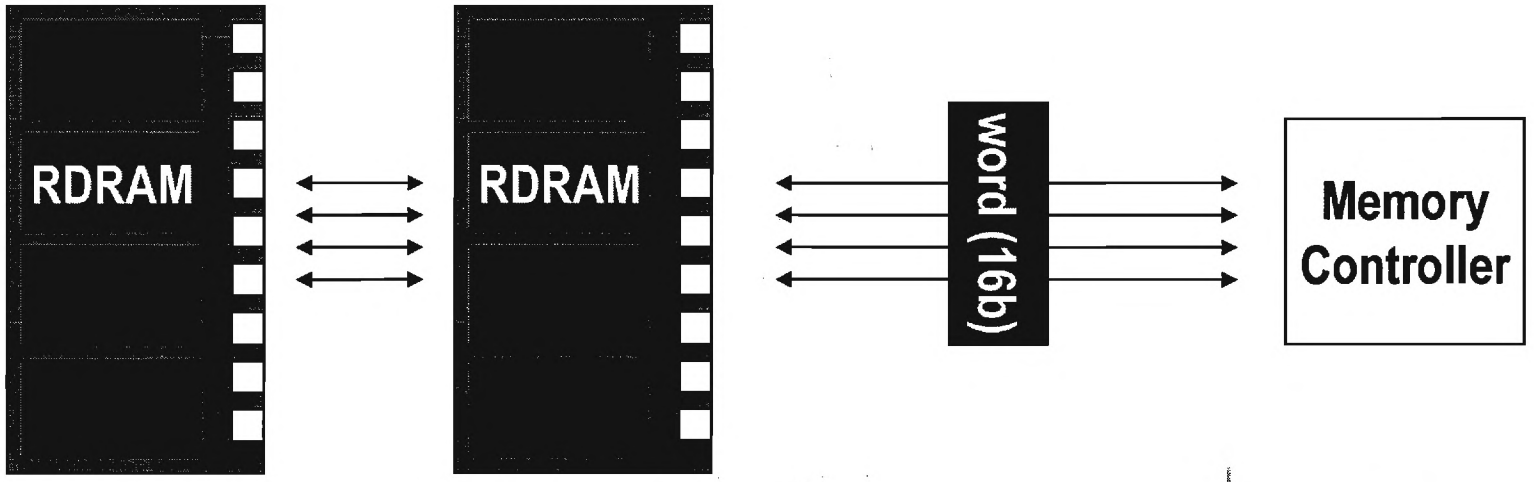
RDRAM

- RDRAM : Rambus DRAM
- Innovation : Reduce crosstalk problems by clocking a narrower memory channel faster.
- RDRAM only transfers 16bits at once.



RDRAM

- RDRAM : Rambus DRAM
- Innovation : Reduce crosstalk problems by clocking a narrower memory channel faster.
- RDRAM is also point to point, not a bus.



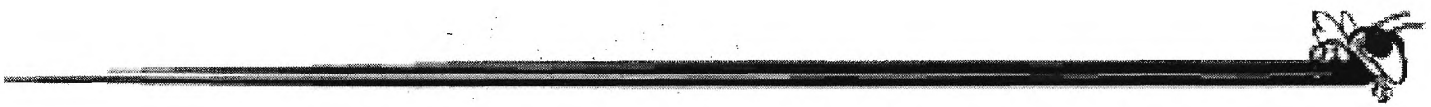
RDRAM

- By only transferring 16 bits and being point-to-point:
  - Only need to lay out 16 data wires on motherboard
  - Can clock much higher since less interference
    - RDRAM clocks at 400-533 Mhz
- RDRAM also uses double-data rate
- Rambus sets rating system:
  - PC1066: 533 Mcyc/sec * 2 trans/cyc * 2 B/trans = **2133 MB/sec**
  - Rating is "PC" + the Mtransfers/sec
  - Often >1 channels are used (more wires, but not too bad)

RDRAM

- RDRAM had many problems holding it back
  - Adding modules lengthens memory channel and thus adds latency.
    - Latency was already hurting (R: 60ns, S: 33ns)
  - Each module must be able to talk to the next one
    - Adds logic to memory device (expensive)
    - Memory is low margin business
  - Wasn't that much faster than DDR
  - Most profit made from Rambus's patents
    - Suing other companies making memory
  - Intel was only supporter. And not any more.

XDR



- XDR DRAM : ??? (Next generation Rambus DRAM)
- Innovation : Octal data rate and LVDS.
- Eight transactions per cycle!
- Instead of using a signal and ground pin per bit, use two wires as a transmission line
  - Much lower voltage needed (+/- 0.2V)
- 400 Mcyc/sec * 8 trans/cyc * 2 B/trans = **6400 MB/sec**
- Future unclear: Rambus says its targeting "consumer electronics" and "graphics applications", but no one has licensed the technology…

DDR2

- Second iteration of DDR standard
- Innovation : Bursts transactions 4 per cycle.
  - Effectively gives QDR performance
  - Internal frequency is not doubled or quaded though.
  - Lower external frequency than DDR1.
- Also lowers I/O from 2.5 to 1.8V.
- Standard set by JEDEC:
  - PC2-4300 : 133 Mcyc/sec * 4 trans/cyc * 8 B/trans = **4266 MB/sec**
- Same bandwidth as best DDR1 (but should scale)
- Still a lot of wires…

GDDR3

- GDDR3 : Graphics DDR (3rd revision of DDR)
- Graphics boards are the real demanders of high memory bandwidth today:
  - Nvidia GeforceFX 5950: **30.4 GB/sec**, 256bit wide!!!
  - Also implements hardware compression on the memory bus to pack more bandwidth in.
  - And game performance is still memory bound!
- Innovation : electrical enhancements to allow higher clock speeds (600-800Mhz).
- 800 Mcyc/sec * 2 trans/cyc * 32 B/trans = **51.2 GB/sec**

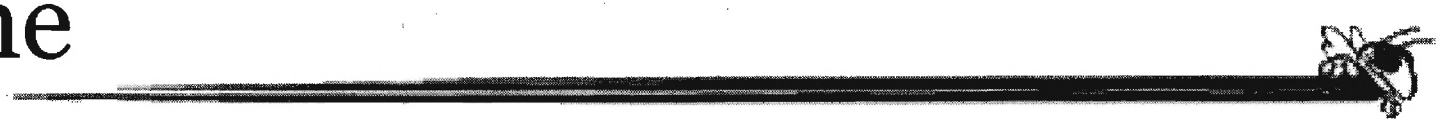
Performance Summary

Variant	Max Clock Rate (Mhz)	Transactions per cycle	Width (bytes)	Peak Bandwidth (GB/sec)
EDO	66	1	4	0.266
SDRAM	133	1	8	1.066
DDR	266	2	8	4.266
RDRAM	533	2	2	2.133
XDR	400	8	2	6.400
DDR2	133	4	8	4.266
GDDR3	800	4	32	51.200

Mini Conclusion

- Specifications, clock rates, and electrical enhancements can be confusing.
- But all of these technologies are all just DRAM with minor enhancements.
  - Same number of transistors per bit.
  - No way to avoid refreshes.
  - Still way slower than SRAM.

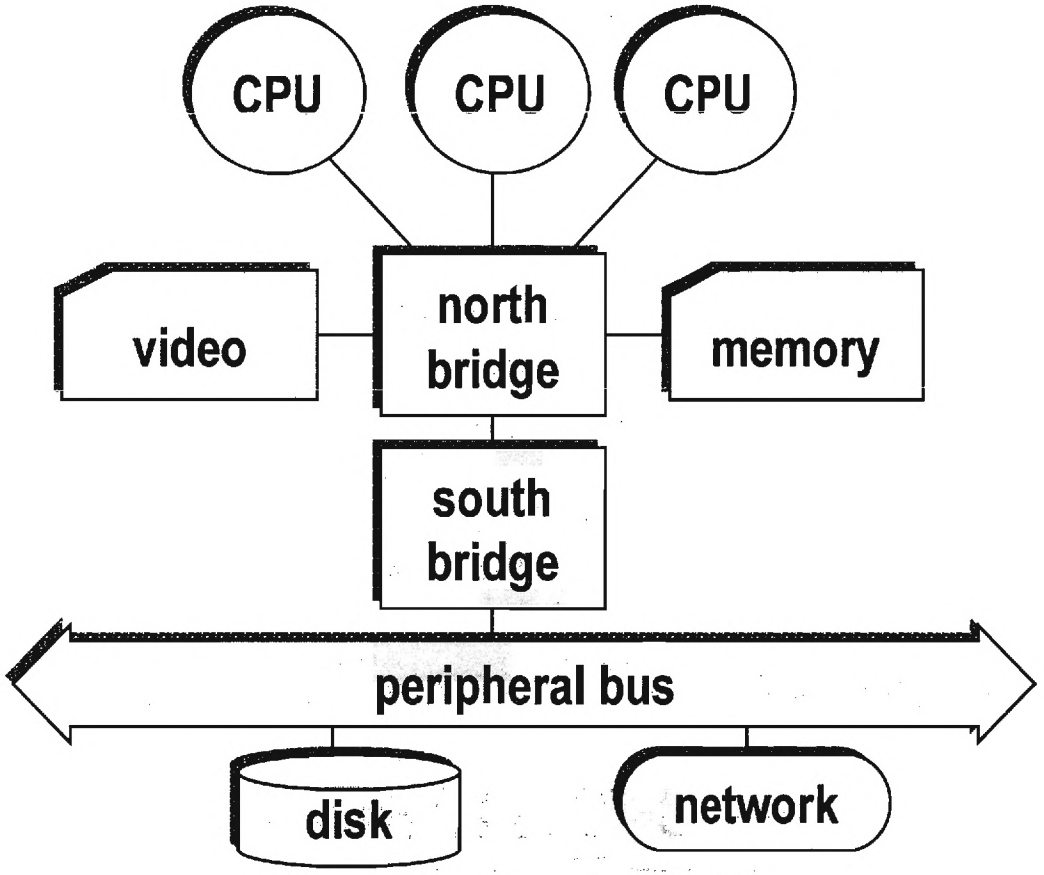
Outline



- Memory Technology
  - SRAM
  - DRAM
- Commercial Offerings
  - FPM DRAM / EDO DRAM
  - SDRAM / DDR SDRAM
  - RDRAM / XDR
  - DDR 2 / GDDR 3
- Northbridge
  - Old role
  - New role

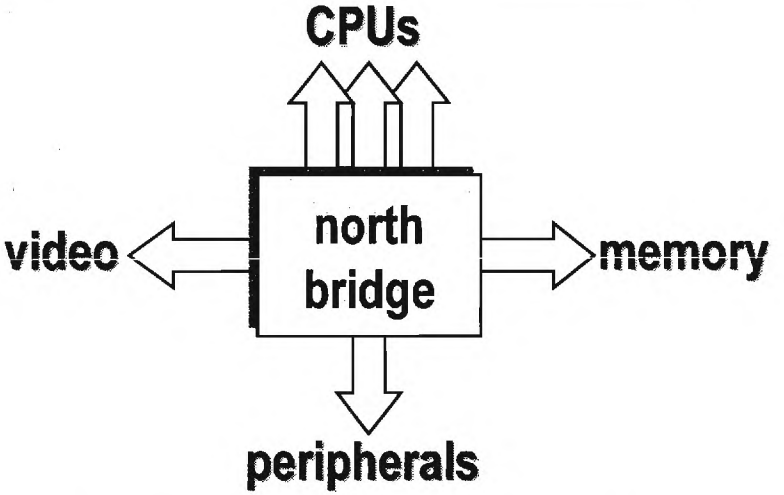
Central Processing Unit?

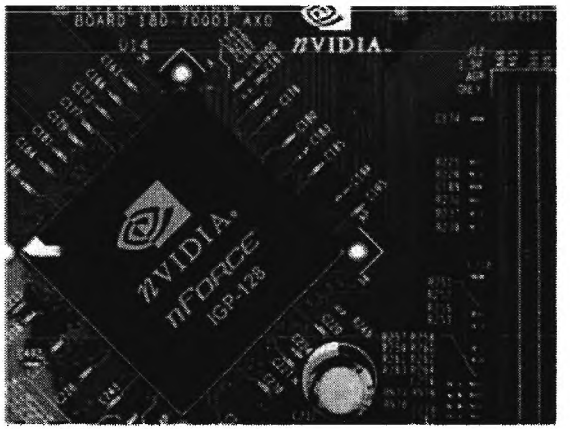
- Central processing unit (CPU) is a misnomer:



Northbridge

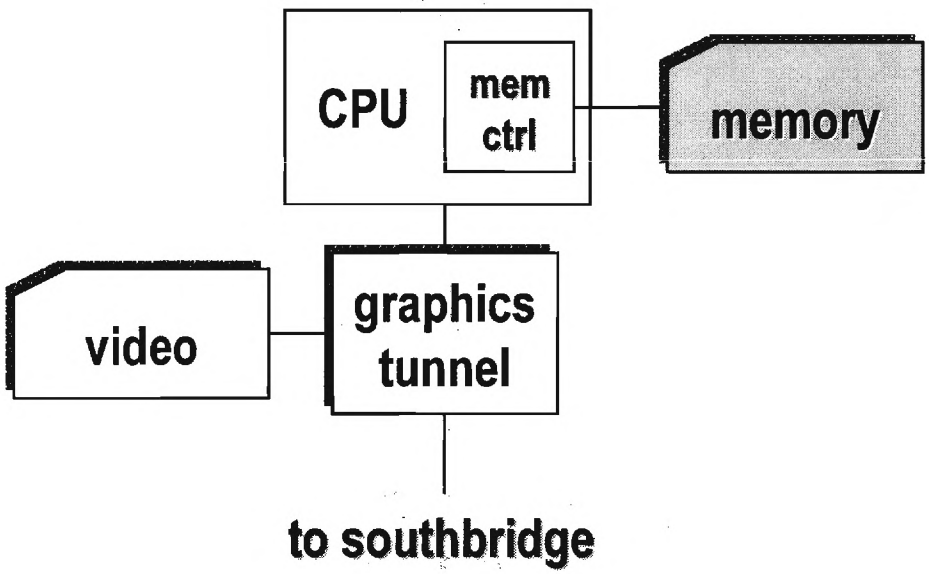
- The Northbridge serves as the crossroads between all the major components in a modern PC.
- Not surprisingly, it is a performance bottleneck.
  - CPU must negotiate for memory access!
- Layout issues
  - Each connection is hundreds of traces… Nforce2 NB has 840 pins (no SMP)!
  - Must be near to CPUs, AGP slot, memory slots and south bridge!
  - Can't run too hot or will interfere with cooling nearby CPU.





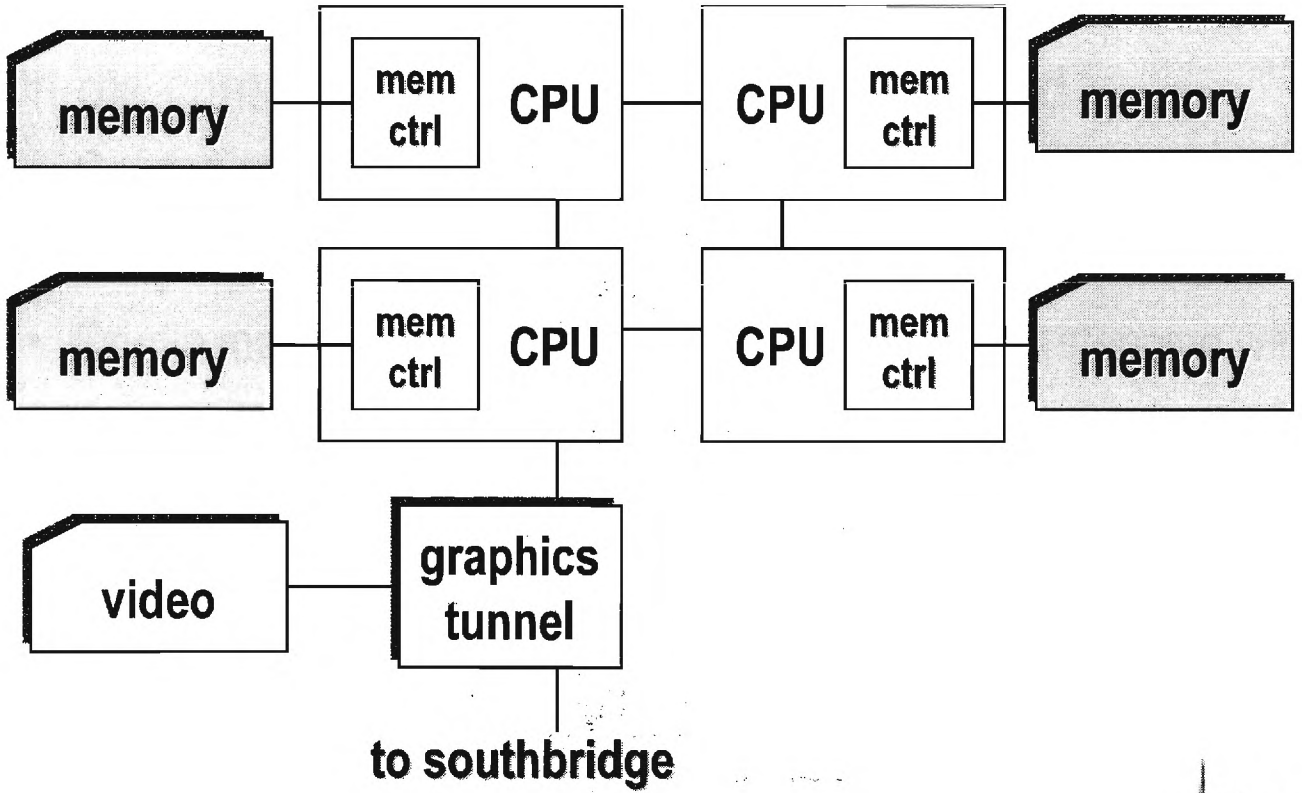
Memory Controller Integration

- Processors such as the AMD Opteron and Transmeta Efficeon integrate the memory controller into the die.
  - The remaining northbridge is called the graphics tunnel.



Point-to-Point MP

- The AMD Opteron also has point-to-point multiprocessor connections.
  - Only one processor connects to the graphics tunnel.
  - Each processor gets their own memory banks (NUMA).



Conclusion

- Memory standards change monthly, but the fundamentals of DRAM haven't changed for decades.
- Legacy infrastructure and device support will keep things that way for a while.
- SRAM and DRAM are complementary
  - speed versus density
- Access to memory is becoming more direct thanks to the northbridge's changing role.